



THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Master of Engineering (Information and Communication Technology for Embedded Systems)

DEGREE

Information and Communication Technology for Embedded Systems

Electrical Engineering

FIELD

DEPARTMENT

TITLE: The SIP-Based Radio over Internet Protocol Interoperability
Communication System For Emergency Situation

NAME: Ms. Sirinya Tabsombat

THIS THESIS HAS BEEN ACCEPTED BY

THESIS ADVISOR

(Associate Professor Mongkol Raksapatcharawong, Ph.D.)

THESIS CO-ADVISOR

(Mr.Teera Phatrapornnant, Ph.D.)

THESIS CO-ADVISOR

(Associate Professor Katsunori Yamaoka, Dr.Eng.)

DEPARTMENT HEAD

(Assistant Professor Srijidtra Charoenlarnopparut, Ph.D.)

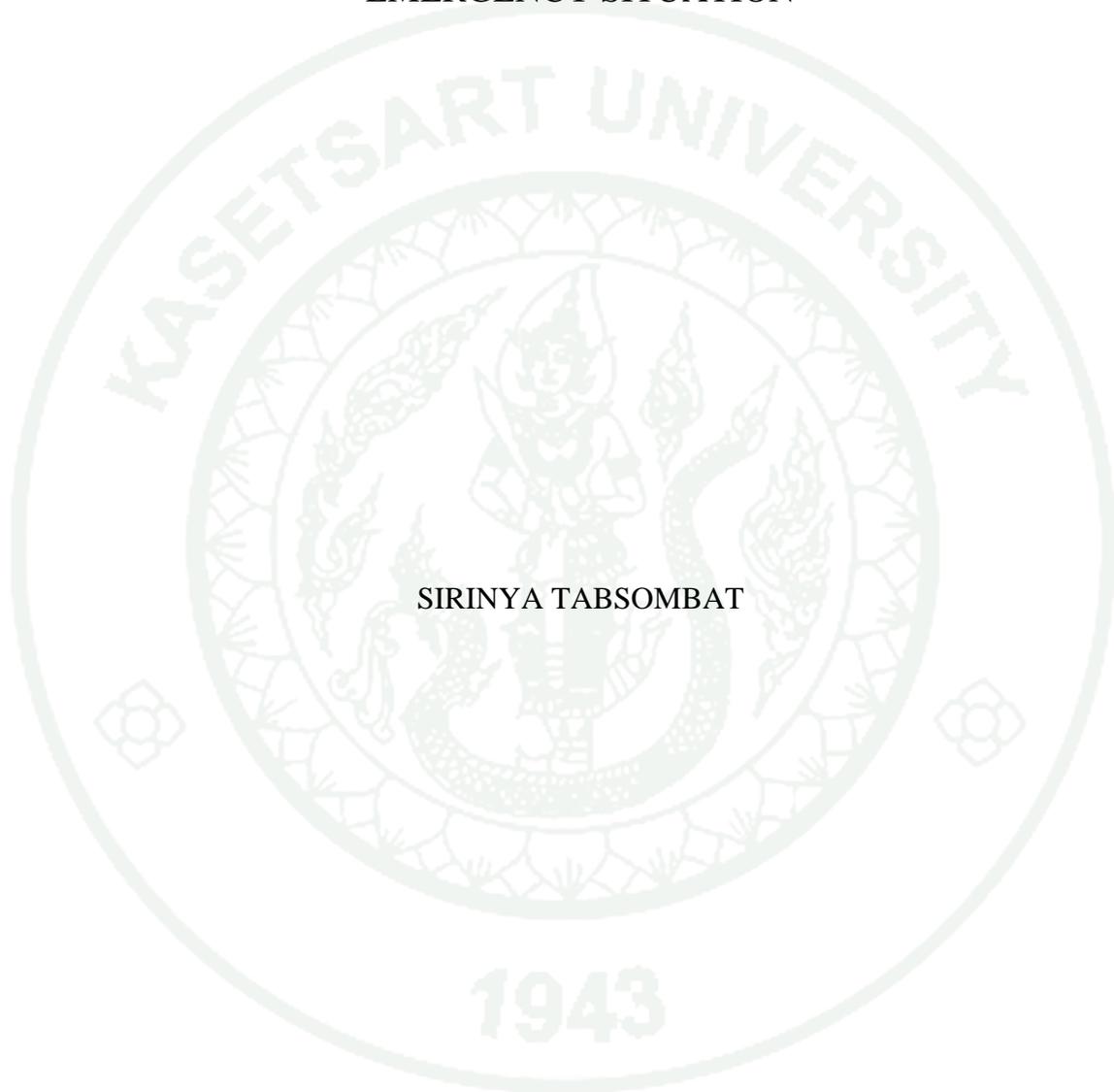
APPROVED BY THE GRADUATE SCHOOL ON _____

DEAN

(Associate Professor Gunjana Theeragool, D.Agr.)

THESIS

THE SIP-BASED RADIO OVER INTERNET PROTOCOL
INTEROPERABILITY COMMUNICATION SYSTEM FOR
EMERGENCY SITUATION



SIRINYA TABSOMBAT

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Master of Engineering (Information and Communication Technology for Embedded Systems)
Graduate School, Kasetsart University
2010

Sirinya Tabsombat 2010: The SIP-Based Radio Over Internet Protocol Interoperability Communication System for Emergency Situation. Master of Engineering (Information and Communication Technology for Embedded Systems), Major Field: Information and Communication Technology for Embedded Systems, Department of Electrical Engineering. Thesis Advisor: Associate Professor Mongkol Raksapatcharawong, Ph.D. 126 pages.

A policy-based network such as a communication system for an emergency situation needs a good reliability. In this work, we present a Linux-based Radio over IP system. The system is a cost-effective IP interoperability communication system that integrated the mobile radio, typical public switched telephone network lines, cell phone, native IP devices etc. All equipment can access the communication with each other via IP network using hardware-software cooperated by asterisk SIP IP-PBX, running under Linux OS.

First, we investigated the current open source technologies to find an appropriate technology mainly based on its stability, ease of use, supported CODECs, and then integrate to our solution. After that, we developed necessary hardware to support effective use of RoIP on existing VoIP technology. And finally, we find an appropriate configuration for emergency use. The main constraints are: satellite link bandwidth, codec selection, jitter buffer setting, and VOX threshold.

After the evaluation of performance tests, we can determine the basic configurations for the system in order to readily for use as an interoperability system to support an emergency situation communication

Student's signature

Thesis Advisor's signature

ACKNOWLEDGEMENTS

I am profoundly indebted to my thesis advisor, Associate Professor Mongkol Raksapatcharawong, Ph.D., for his encouragement and valuable advice. It was his kindness giving me a great opportunity to enter the TAIST Tokyo Tech - ICTES Program, giving me a great degree of freedom to choose a topic and pursue my research. Beside the academic advice, I am also grateful for his caring personality and his great optimistic attitude.

I would sincerely like to thank my thesis co-advisors, Associate Professor Katsunori Yamaoka, Dr.Eng., from Tokyo Institute of Technology, Mr.Teera Phatrapornnant, Ph.D., Mr.Waranyoo Phiwthongkham and Mr.Phisanu Duangtanoo from National Electronics and Computer Technology Center (NECTEC) who always supports and helps me to finish all papers and thesis works.

I would like to express our thanks to TAIST Tokyo Tech - ICTES Program faculties, Thailand Advanced Institute of Science and Technology - Tokyo Institute of Technology (TAIST-Tokyo Tech), National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology (Tokyo Tech) , Kasetsart University (KU) and the Ministry of Information and Communication Technology (MICT) for supporting this research.

Finally, all my attempts and appreciations devotes to my parents, my sister and brother who understand and support me all the time. And deep gratitude goes to my masters, Phra Ajarn Aekachai Issaro Bhikkhu and Kru Chanoknate Khammong who bring me a good spirit and always teach me to do all my work for goodness and for publicity.

Sirinya Tabsombat

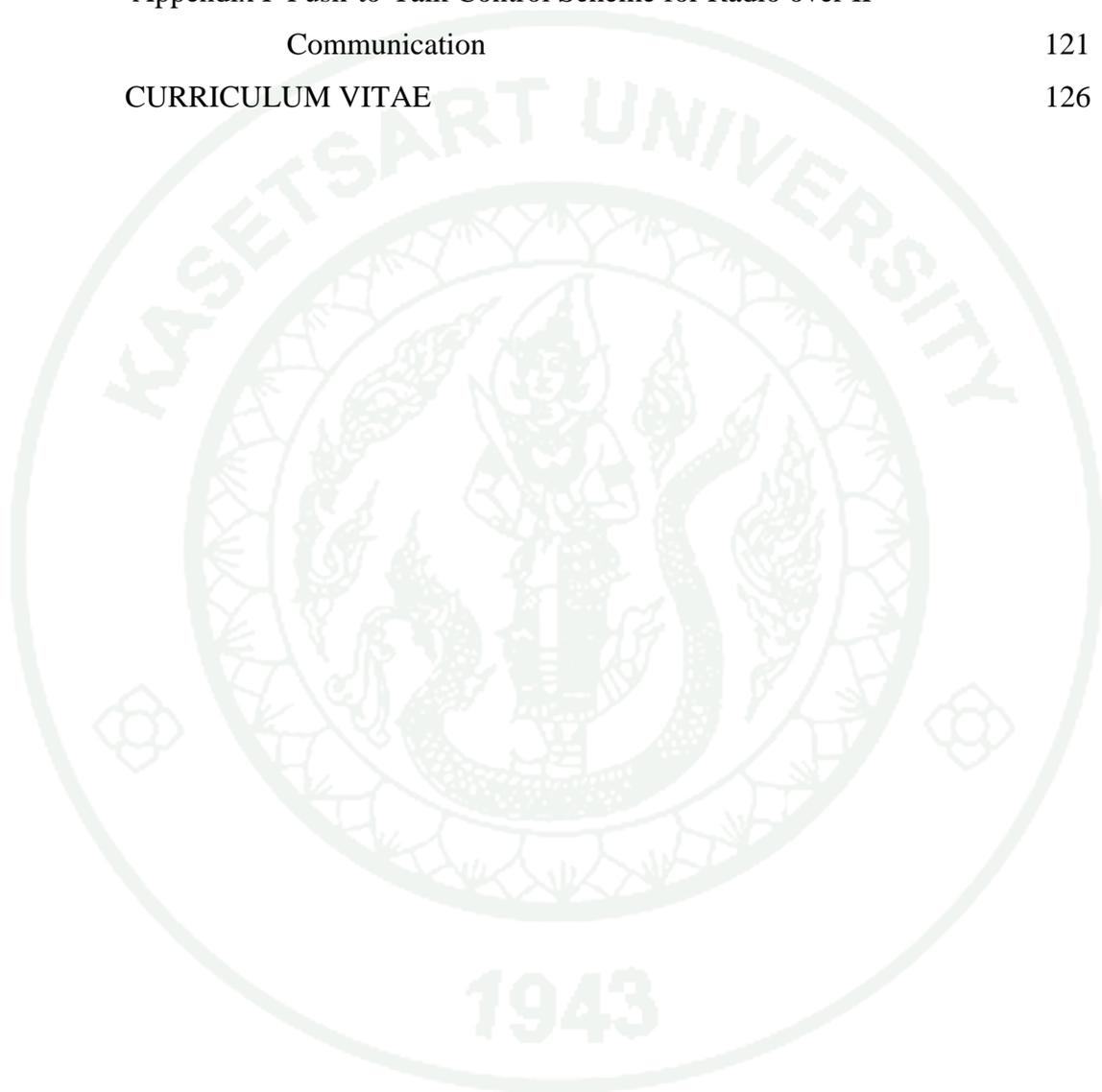
June 2010

TABLE OF CONTENTS

	Page
TABLE OF CONTENTS	i
LIST OF TABLES	ii
LIST OF FIGURES	iii
LIST OF ABBREVIATIONS	viii
INTRODUCTION	1
OBJECTIVES	3
LITERATURE REVIEW	4
MATERIALS AND METHODS	15
Materials	15
Methods	20
RESULTS AND DISCUSSION	69
Results	69
Discussion	81
CONCLUSION AND RECOMMENDATION	82
Conclusion	82
Recommendation	83
LITERATURE CITED	84
APPENDICES	86
Appendix A Supported CODEC and Equipments	87
Appendix B ITU - Telecommunication Standardization Sector Temporary Document XX-E WP 2/12	89
Appendix C .XML Useful Command for Load Test	93
Appendix D Performance Analysis of Asterisk SIP Server for Emergency Communications	104
Appendix E Radio over IP Prototyping: a Communication System for Emergency Response	115

TABLE OF CONTENTS (Continued)

	Page
Appendix F Push-to-Talk Control Scheme for Radio over IP Communication	121
CURRICULUM VITAE	126



LIST OF TABLES

Table		Page
1	Comparison between H.323 and SIP Protocol	8
2	Comparison of available emergency interoperability communication systems	10
3	Mean Opinion Score (MOS) according to ITU-T standard	13
4	Example of codec and MOS relationship	14
5	Comparison between VoIP Solutions	26
6	End-to-End Delay from Push-to-Talk	37
7	G.711 and G.729 Codec Information	54
8	Bandwidth Calculations for each codec	54
9	Some Characteristics of G.711 codec	60
10	Voice Transmission via 2/1 Mbps IPstar Broadband satellite link using G.711 CODEC and SIP Trunk, with VoIP Phone	69
11	Voice Transmission via 2/1 Mbps IPstar Broadband satellite link using G.729 CODEC and SIP Trunk, with VoIP Phone	71
12	Video Transmission via 2/1 Mbps IPstar Broadband satellite link using H.263 CODEC and SIP Trunk (software: eyebeam 1.5)	72
13	Characteristics of voice transmission based on terminal types and CODECs	76

LIST OF FIGURES

Figure		Page
1	Block Diagram of VoIP.	4
2	Block Diagram of RoIP.	5
3	Frequency ranges of mobile radio communication.	7
4	The connection between SIP servers.	9
5	Asterisk™ Architecture.	12
6	Effect of Packet loss to MOS.	14
7	Server PC.	15
8	Client PC.	15
9	Linux Cent OS 5.0 kernel 2.6.18-164.	16
10	Asterisk IP-PBX Software.	16
11	VoIP Gateway and Router.	17
12	RJ45, RJ11.	17
13	X-Lite Softphone.	17
14	VoIP Phone.	18
15	Analog Phone.	18
16	Mobile Radio.	18
17	WiFi equipments.	19
18	Server Box.	19
19	UPS.	19
20	Asterisk Control Panel.	21
21	Tribox Control Panel.	22
22	Elastix Control Panel.	23
23	Slast Control Panel.	23
24	Askosia Control Panel.	24
25	AsteriskWin32 Control Panel.	25

LIST OF FIGURES (Continued)

Figure	Page	
26	FreePBX Control Panel.	25
27	Block diagram of Telephone to Radio controller circuit.	27
28	A Telephone Hybrid Circuit.	28
29	A completed assemble of Telephone Hybrid Circuit.	28
30	The diagram of Telephone Line Detection Circuit.	29
31	Example of telephone line diagram.	29
32	The diagram of Voice Detection Circuit	30
33	The diagram of Attenuator Circuit	31
34	MCU Unit Diagram	32
35	Radio-Phone cooperation via Telephone Hybrid circuit	34
36	Signaling diagram of PTT control scheme	35
37	Single side of Push-to-Talk control scheme	36
38	System Architecture	38
39	The diagram of Software Installation and System Configuration	39
40	The diagram of network connection via satellite	50
41	The Web-based GUI	51
42	Set up a call and video transmission to evaluate the call functions and analyze the Voice quality, Delay, Jitter and Bandwidth used	52
43	Performance Test of voice and video transmission via IPStar Broadband Internet	52
44	The field test of the RoIP interoperability system interconnection between two network bases via satellite	53
45	Bandwidth Calculations per 1 call using SIP Trunk, G.711 codec	55
46	Bandwidth Calculations per 1 call using IAX2 Trunk, G.711 codec	56
47	Bandwidth Calculations per 1 call using SIP Trunk, G.729codec	56
48	Bandwidth Calculations per 1 call using IAX2 Trunk, G.729 codec	57

LIST OF FIGURES (Continued)

Figure		Page
49	Bandwidth Calculations per 1 call using SIP Trunk, G.711 and G.729 transcoder	57
50	Bandwidth Calculations per 1 call using IAX2 Trunk, G.711 and G.729 transcoder	58
51	Experiment for system's response	59
52	Traffic Monitoring by Wireshark	68
53	The statistics graph from the voice transmission via 2/1 Mbps IPstar Broadband satellite link using G.711 CODEC and SIP Trunk	70
54	The statistics graph from the voice transmission via 2/1 Mbps IPstar Broadband satellite link using G.729 CODEC and SIP Trunk	71
55	The voice transmission by VoIP Phone using G.711	73
56	The voice transmission by VoIP Phone using G.711 codec	73
57	The video transmission by SoftPhone using G.711 and H.263 codec	73
58	The video transmission by SoftPhone using G.711 and H.263	74
59	The voice transmission by VoIP Phone using G.729	74
60	The voice transmission by VoIP Phone using G.729	74
61	The video transmission by SoftPhone using G.729 and H.263 codec	75
62	The video transmission by SoftPhone using G.729 and H.263 codec	75
63	The video transmission by SoftPhone using G.711 and H.263 codec (no jitter buffer)	77
64	The video transmission by SoftPhone using G.711 and H.263 codec with Jitterbuffer configuration	78

LIST OF FIGURES (Continued)

Figure		Page
65	The video transmission by SoftPhone using G.729 and H.263 codec (no jitter buffer)	78
66	The video transmission by SoftPhone using G.729 and H.263 codec with Jitterbuffer configuration	78
67	Call Retransmissions and Call Fail from voice traffic simulations	79
68	Call Retransmissions and Call Fail from video traffic simulations	80

LIST OF ABBREVIATIONS

A/D	=	Analog-to- Digital conversion
AM	=	Amplitude Modulation
ATA	=	Analog Telephony Adapter
CB	=	Citizen Band (245 MHz)
CID	=	Caller Identification
COR	=	Carrier Operated Relay
D/A	=	Digital / Analog conversion
DID	=	Direct-Inward-Dial
DDNS	=	Dynamic Domain Name Server
ETSI	=	European Telecommunications Standards Institute
GUI	=	Graphic User Interface
HF	=	High frequency
IAX	=	Inter-Asterisk eXchange protocol
IEEE	=	Institute of Electrical and Electronics Engineers
IP	=	Internet Protocol
ITU	=	International Telecommunications Union
LAN	=	Local Area Network
LF	=	Low frequency
MF	=	Medium frequency
MOS	=	Mean Opinion Score
OSI	=	Open Systems Interconnection
PBX	=	Private Box Exchange
PPP	=	Point-to-Point Protocol
PSTN	=	Public Switched Telephone Network
PT	=	Payload Type
PTT	=	Push-To-Talk
QoS	=	Quality of Service
RFC	=	Request For Comments (standards)
RoIP	=	Radio Over IP
RTP	=	Real-time Transport Protocol

LIST OF ABBREVIATIONS (Continued)

SHF	=	Super high frequency
SIP	=	Session Initiation Protocol
SNR	=	Signal-to-Noise Ratio
SSB	=	Single Sideband
TCP	=	Transmission Control Protocol
UDP	=	User Datagram Protocol
UHF	=	Ultra high frequency
VHF	=	Very high frequency
VoIP	=	Voice Over Internet Protocol
VPN	=	Virtual Private Network
InterOp	=	InterOperability System

THE SIP-BASED RADIO OVER INTERNET PROTOCOL INTEROPERABILITY COMMUNICATION SYSTEM FOR EMERGENCY SITUATION

INTRODUCTION

In crisis and emergency situation, such as tsunami, flooding, or other kind of disasters, a backup communication system is very important to inform the location and other critical information such as the impacted area, the nature of impact, and the locations of key assets (e.g., shelters, disaster equipment and potential responders). Appropriate decisions and actions cannot be made without this information and the tools necessary to share and collaborate on the crisis. Imagine the emergency rescue mission during the broken link of telephone systems or lack of the connection. In this case, the first communication tool we normally use is a mobile radio, as it needs only the carrier frequency to keep the communication connected.

According to the device's properties, mobile radio has several individual bandwidth-range in use, and the accessibility between the different channels is forbidden. In the real situation, a number of mobile radios available in the same area during the crisis are operating on different channels and are assigned to different group of users. The lack of consistent policy for collaboration, together with protocols and structures for coordination and communication, has been a challenge. We've looked for a tool that enables emergency response agencies to exchange critical communications and data with one another, permitting them to work together effectively. The solution is to find an interoperability network that allows various mobile radio systems—such as UHF, VHF, 245 MHz—to communicate with each other during emergency situations.

The network core service of focus is the Voice-Over IP (VoIP) system because it allows information from different sources to communicate on the IP network. For example, voice signals from radio or telephone systems are digitized by the VoIP gateway, loaded onto the IP network and then transported to wherever the packets are

destined. The system uses the IP and SIP protocol standards, running on Linux platform, connecting with Mobile Radio transmitter and receiver units, IP and non-IP wired, wireless, and satellite broadband networks.

During disaster, any emergency mobile radio can access the communication and meet its mission needs in time. These systems can be linked, monitored and controlled over an IP network. Where system's requirements are (Health Canada, 2003):

1. The system must be readily deployed in response to critical situation: any communication devices can access together immediately.
2. The system can support a number of incoming calls sufficient to handle a critical situation.
3. The system performance would not be severely degraded by a large amount of packet loss.

Our system represents one of the interoperable solutions which enable the communication between any heterogeneous communication devices via our RoIP network. With open system development, the Radio over IP interoperability system is the first non-commercial RoIP system in Thailand that meets the cost-effective solution and can best provide the functionality and scalability necessary to meet the emergency communication solutions.

OBJECTIVES

Software and hardware that works together could unify different forms of communications devices. Recently, several mobile devices are already “VoIP-enabled” allowing a mobile user to use VoIP services when WLAN coverage allows it (C. Castro *et al.*, 2007).

In this approach, we developed the interoperability communication system under the specific policy for emergency situation. The solutions for our need are:

1. Investigate the current open source technologies to find an appropriate technology mainly based on its stability, ease of use, supported CODECs, and then integrate to our solution.
2. Develop necessary hardware to support effective use of RoIP on existing VoIP technology.
3. Find an appropriate configuration for emergency use. The main constraints are: satellite link bandwidth, codec selection, jitter buffer setting, and VOX threshold.

LITERATURE REVIEW

1. Voice over IP: VoIP

Internet telephony refers to communications services — voice, facsimile, and/or voice-messaging applications — that are transported via the Internet, rather than the public switched telephone network (PSTN). The basic steps involved in originating an Internet telephone call are conversion of the analog voice signal to digital format and compression/translation of the signal into Internet protocol (IP) packets for transmission over the Internet; the process is reversed at the receiving end.

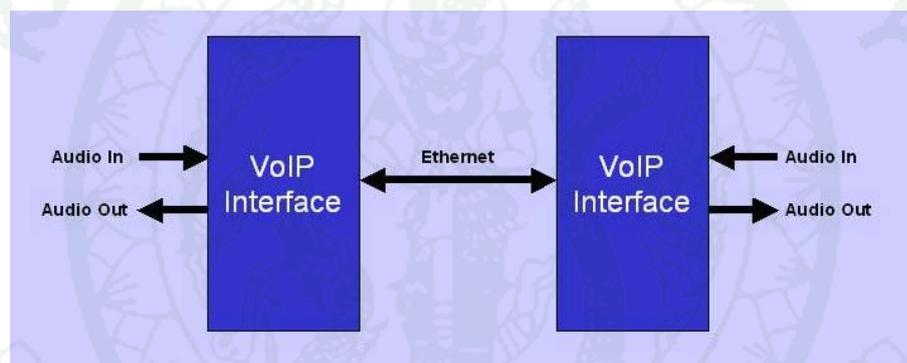


Figure 1 Block Diagram of VoIP

VOIP systems employ session control protocols to control the set-up and tear-down of calls as well as audio codecs which encode speech allowing transmission over an IP network as digital audio via an audio stream. Codec use is varied between different implementations of VOIP (and often a range of codecs are used); some implementations rely on narrowband and compressed speech, while others support high fidelity stereo codecs.

2. Radio over IP: RoIP

Radio over IP is an upper level technology of the more common term, VoIP (Voice over Internet Protocol). VoIP provides the Internet vehicle for moving voice audio from point-to-point via the Internet. There are many low cost providers of VoIP, as the only requirement is to carry voice from point-to-point. See Figure 2: RoIP Block Diagram below (Raytheon, White Paper: WP-3004-2).

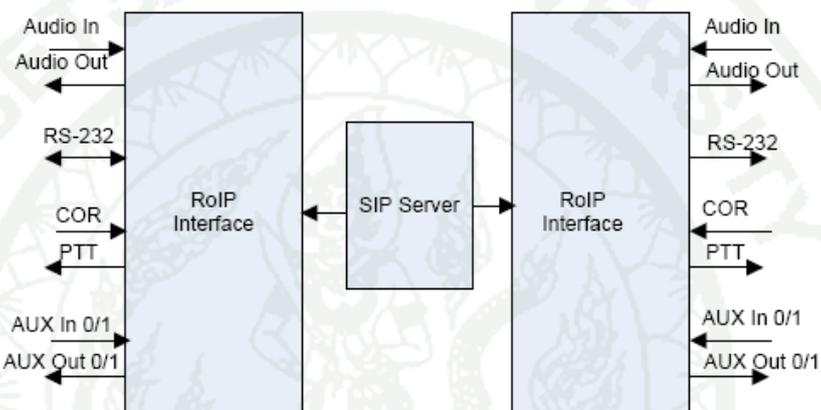


Figure 2 Block Diagram of RoIP

There are two signals required to control devices and send audio to and from radios. The two signals are PTT (Push-To-Talk) and COR (Carrier Operated Relay.)

PTT - This function is required by a radio when the user wishes for the audio to travel out via the radio and over the air to a distant radio user. The user pushing the PTT button on the side of the radio microphone to speak achieves PTT on a hand held radio. The button push keys the radio and allows the voice audio to travel via radio waves to the distant end-user radio. Any time voice audio travels from a radio over the air, that radio must receive the PTT to allow transmission of the audio. All radios have either a PTT button on the microphone, or a PTT input on its input/output connectors for external equipment usage. (Raytheon, White Paper: WP-3004-2)

COR- This function allows a device connected to a radio, to have prior knowledge the receive signal is coming from the radio. Most radios do not have a COR output. When a COR output is not available, the device connected to the radio must have the ability to create the COR. Creating COR for a radio, which does not supply COR, can be done by using VOX (Voice Operated Xmit) detection. The device receiving the radios voice signal must use VOX and audio delay to successfully key and forward the audio. (Raytheon, White Paper: WP-3004-2)

VOX - The VOX algorithm will signal COR present whenever the incoming audio exceeds a set threshold. The signal can be tones, voice or noise, and the threshold can be varied for different conditions. (Raytheon, White Paper: WP-3004-2)

These two requirements, PTT and COR, are a step above the standard VOIP technologies; it requires a technology capable of providing a means to transport PTT and COR, while traveling the Internet with the VOIP, or voice audio. (Raytheon, White Paper: WP-3004-2).

The more advantage when implementing to Radio over IP is to enable the difference channels of radio frequencies to be accessible. In generally, the frequency ranges of mobile radio communication are divided to HF, UHF, and VHF. The radio over IP system will change the form of analog communication to digital. Therefore, all radio channels will access together as an IP devices.

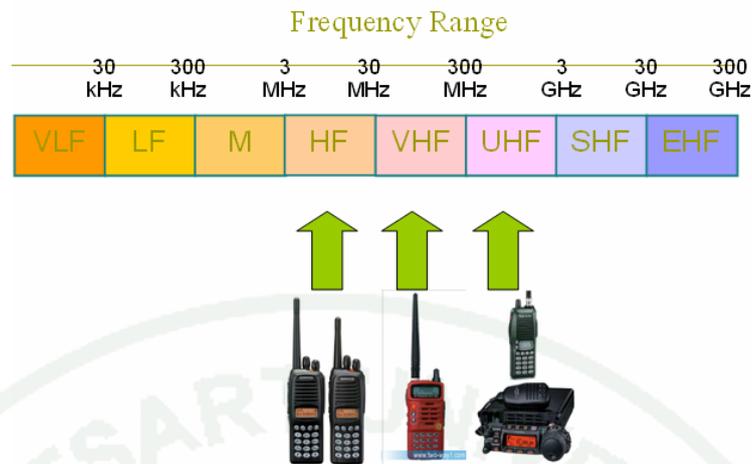


Figure 3 Frequency ranges of mobile radio communication

3. Protocols for Voice over IP

H.323: The first generation VOIP Protocol, high complexity and low scalability.

MGCP: Media Gateway Control Protocol, New protocol, simple

SIP: Session Initial Protocol, the most popular protocol, service oriented, been used widely (Standards Track RFC 3261 SIP, 2002).

Megaco/H.248: is an implementation of the Media Gateway Control Protocol architecture for controlling Media Gateways on Internet Protocol (IP) networks and the public switched telephone network (PSTN). However, it is a different protocol and the underlying differences make them incompatible.

In opened system development, in order to compatible with general available devices and easily to upgrade or make a configuration change, we need to select the popular protocol. The two most popular protocols for Voice over IP are H.323 and SIP. H.323 is more complex and unsuitable for open system (difficult to upgrade or

change the configuration). SIP has more simple structure and more scalable. It also consume less resources e.g. bandwidth consumption, memory consumption.

In Table 1, the comparison between H.323 and SIP Protocol is shown.

Table 1 Comparison between H.323 and SIP Protocol

H.323	SIP
Complex Protocol	Comparatively Simpler
Binary representation for its message	Textual representation
Not very modular	Very modular
Not vary scalable	Highly scalable
Complex Signaling	Simple Signaling
Hundreds of Header	37 Headers
Loop Detection is difficult	Loop Detection is comparatively easy

The Session Initiation Protocol (SIP) is a text-based protocol that was originally designed by Henning Schulzrinne and Mark Handley starting in 1996. The SIP protocol is a TCP/IP-based Application Layer protocol, with syntax similar to that of HTTP. SIP typically uses port number 5060 or 5061 both for UDP and TCP, and it may implement other transport protocols. A motivating goal for SIP was to provide a signaling and call setup protocol for IP-based communications. SIP also defines server network elements. It has been standardized and governed primarily by the IETF, while other protocols, such as H.323, have been traditionally associated with the International Telecommunication Union (ITU).

Two SIP endpoints can communicate without any intervening SIP infrastructure, because there are various types of gateways compatible at the edge between a SIP network and other networks which is why the protocol is described as peer-to-peer. This approach is often impractical for a typical public service telephone

network, therefore the advantages of simplify, expansibility and modularization can provides the possibility of interoperability communication system implementation.



Figure 4 The connection between SIP servers

There are some similarities between our system and others. But the difference is the capabilities in:

3.1 System configuration

Since it is the open system, all configurations and all changes are enabled. For major section e.g. system core (main server PC, OS), service core (main IP-PBX software) and system policy (codec allowing, packet loss control, network connection method etc.). And for minor section e.g. gateways and communication devices configuration.

3.2 System control

From 2.1, each site is independent in system control and configuration policy based on current environment.

3.3 Readily for 2 Mbps satellite link.

Since the system is objective for emergency communication, the prior configuration is based on the low bandwidth and resources.

Table 2 Comparison of available emergency interoperability communication systems

Properties	Our System	NECTEC system	Provincial Administration Mobile Car	Ministry of Interior. Mobile Car	Crime Investigation Division Mobile Car	USA international Red-Cross	CISCO System
1. Link Method	Interoperation	Interoperation	Interoperation	Satellite	Satellite	Interoperation	Interoperation
2. Covered Area	30km- Radio 3km- WiFi IPStar Broadband Internet	2km-WiFi IPStar Broadband Internet	30km-Radio Internet via Satellite	Internet Via Satellite	Internet Via Satellite	?	?
3. System Architecture	Open / SIP	Closed / SIP	Closed / N/A	Closed / N/A	Closed / N/A	Closed / N/A	Closed / H.323
4. Services							
4.1. Radio	✓	✓	✓	N/A	N/A	✓	✓
4.2. Wi-Fi	✓	✓	N/A	N/A	N/A	✓	✓
4.3. PSTN (Public Switch Telephone Network)	✓	✓	✓	N/A	N/A	✓	✓
4.4. MSC (Mobile Switching Center)	✓	✓	✓	N/A	N/A	✓	✓
4.5. Internet Via Satellite	✓	✓	✓	✓	✓	✓	✓
4.6. IP Phone	✓	✓	N/A	N/A	N/A	✓	✓
5. Cross frequency for Radio	✓	✓	✓	N/A	N/A	✓	✓
6. Concurrent call Capability	150- internal call Immediately 10- internal call Immediately	25- internal call Immediately	10- internal call Immediately 2- internal call Immediately	N/A	N/A	10- internal call Immediately 2- internal call Immediately	Depends on the System
7. Uninterruptible Power Supply	✓	✓	✓	✓	✓	✓	
8. System Control	On-site	On-site	On-site	On-site	On-site	On-site	Control Center

Table 2 (Continued) Comparison of available emergency interoperability communication systems

Properties	Our System	NECTEC system	Provincial Administration Mobile Car	Ministry of Interior. Mobile Car	Crime Investigation Division Mobile Car	USA international Red-Cross	CISCO System
9. Bandwidth (DL/UL)	2Mbps/1Mbps	?	2Mbps/1Mbps	2Mbps/1Mbps	2Mbps/1Mbps	?	?
10. Tele-Conference Support	✓ (Video/Voice)	✓ (Voice)	✓ (Voice)	✓ (Video/Voice)	✓ (Video/Voice)	✓ (Video/Voice)	✓ (Video/Voice)
11. Scalability	✓	✓	✓	✓	✓	✓	✓

4. IP-PBX Application

In considering the deployment of VoIP in the contact center may also be thinking about implementing of an Internet Protocol Private Branch Exchange (IP-PBX). An IP-PBX enables connections among the internal telephones of a private organization by allowing calls to be sent via data packets over a data network instead of the traditional phone network. The advantage of an IP-PBX concept is utilized for our system in enable the incorporation between various communication devices.

The IP-PBX Software selected for this research is AsteriskNOW, one of Asterisk PBX Software. The capacity of AsteriskNOW compared with other IP-PBX software are:

1.1 User Friendly: Since it compatible use with FreePBX – the most comfortable GUI Panel.

1.2 The AsteriskNOW is one of stable version of Asterisk that running on LINUX Cent-OS.

AsteriskNOW version 1.6.2.0, FreePBX version 2.5.1.0. An open source software and is available free to all under the GNU General Public License.

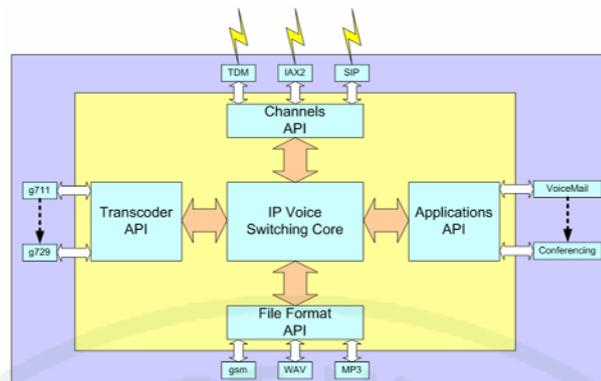


Figure 5 Asterisk™ Architecture

Asterisk is an IP-PBX (Internet Protocol - Private Branch Exchange) software that has many flexibility features for Voice over IP communication via SIP, IAX2 and TDM protocols. It provides various codec supports including with transcoder function. With IP-PBX concept, the general service function such as VoiceMail, Conferencing etc. are created for readily for use.

The interface between Asterisk system and users may be enabled via Graphic User Interface (GUI). The most effective GUI for AsteriskNOW is FreePBX Software that allow user to have basic configure via graphic interface. However, using graphic screen also lead the computer to consume a high percent of resource such as CPUs and RAM.

5. Service Quality and Voice Quality

Packet-switched networks are often used to provide interactive multimedia communications, including real-time voice, video and data services. Packet networks were not designed for real-time multimedia applications such as voice or video communications as they do not provide a dedicated end-to-end connection as is provided by circuit switched networks (M. White *et al.*, 2004).

The main complaint regarding VoIP has a lower level of voice quality than regular telephone technology. Therefore VoIP requires a large amount of data to be compressed and transmitted, then uncompressed and delivered, all in a relatively small amount of time. The policy in develop this VoIP interoperability system expects the system has immediate response. The callers require good voice quality such as low echo, delay or packet loss.

5.1 Service quality evaluation

We will determine the communication over limited bandwidth link (2/1 Mbps of IPstar Satellite), therefore, in case of emergency communication, the system can be readily deployed to support the broadband satellite link or the limited resource that available.

5.2 Voice quality evaluation

We will consider time delay and percent of packet loss. The Mean Opinion Score (MOS) (ITU -Telecommunication Standardization Sector Temporary Document XX-E WP 2/12, 2002) (Q. Walker, 2005) is also advantage in consider the appropriate codec selection for the system.

The MOS is generated by averaging the results of a set of standard (See Appendix B), subjective tests where a number of listeners rate the heard audio quality of test sentences read aloud by both male and female speakers over the communications medium being tested. A listener is required to give each sentence a rating using the following rating scheme:

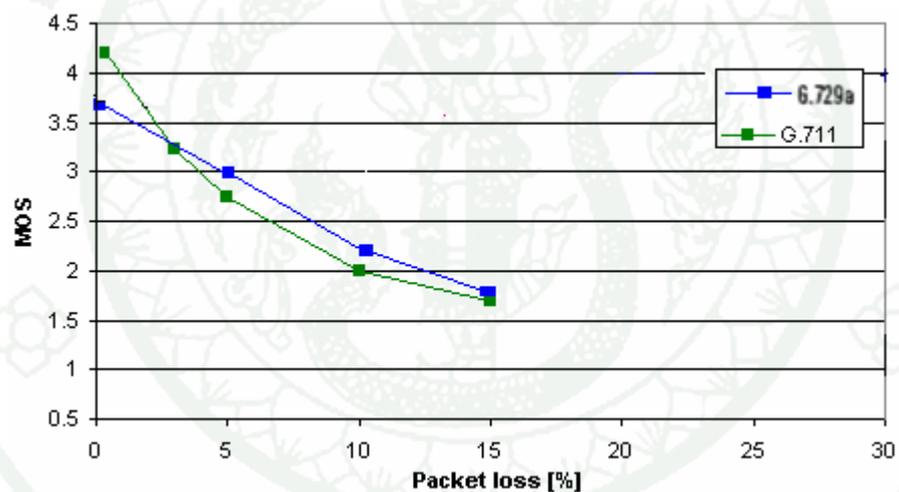
Table 3 Mean Opinion Score (MOS) according to ITU-T standard

MOS	Quality	Characteristics
5	Excellent	Imperceptible
4	Good	Perceptible
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

Table 4 Example of codec and MOS relationship

CODEC	Data Rate	MOS
G.711 (ISDN)	64 kbps	4.3
iLBC	15.2 kbps	4.14
AMR	12.2 kbps	4.14
GSM EFR	12.2 kbps	3.8
G.726 ADPCM	32 kbps	3.8
G.729	8 kbps	3.92
GSM FR	12.2 kbps	3.5

However, the voice quality can be degraded by the packet loss. The more packet loss occurred in the system, the less voice quality is.

**Figure 6** Effect of Packet loss to MOS

MATERIALS AND METHODS

Materials

1. Computer



Figure 7 Server PC

The Intel® Server SR1530SH ,Processor : 64-bit Multi-Core Intel® Xeon® 3200/3000 sequence 1.33 GHz, 8 GB RAM.



Figure 8 Client PC

2. Open source OS



Figure 9 Linux Cent OS 5.0 kernel 2.6.18-164

3. Open source IP-PBX Application (AsteriskNOW)



Figure 10 Asterisk IP-PBX Software

Asterisk version 1.6.2.0, FreePBX version 2.5.1.0: An open source software and is available free to all under the GNU General Public License. It includes many features available in proprietary PBX systems, supporting a mix of traditional and VoIP telephony services.

4. Network Ethernet devices (eg. Switch, Gateway, ATA adapter eg.)



Figure 11 VoIP Gateway and Router

5. Ethernet cables, RJ45, RJ11



Figure 12 RJ45, RJ11

6. SoftPhone Application (eg. X-Lite ,3CX)



Figure 13 X-Lite Softphone

For example, X-Lite version 3.0. X-Lite is developed by CounterPath, available on <http://www.counterpath.com>.

7. VoIP Phone



Figure 14 VoIP Phone

8. Analog Phone



Figure 15 Analog Phone

9. Mobile Radio and equipments



Figure 16 Mobile Radio

10. WiFi equipments



Figure 17 WiFi equipments

11. Server Box



Figure 18 Server Box

12. UPS



Figure 19 UPS

Methods

1. We will first investigate existing VoIP open source solutions that meet our criteria which are stability, ease of use, and supported CODECs.

After this section, should add a table to compare existing solution in terms of stability, ease of use, CODEC support, and Operating System supported.

2. Concurrently we will develop hardware deemed necessary to support effective use of RoIP on existing VoIP technology.

3. After deliberate on the open source solutions, we will choose one to be demonstrated with our system to find an appropriate configuration under emergency communication. Our assumption is to utilize an IP-Star broadband internet service at 2/1Mbps (Downlink/Uplink). The appropriate configuration is based on codec selection, jitter buffer setting, and VOX threshold which leads to desired number of simultaneous calls, voice and video quality in terms of packet loss, end-to-end delay, and jitter delay.

1. Investigating existing VoIP solutions

There are several types of Voice over IP Solutions in both commercial and free software that support some basic tools for users. But for developing to meet a custom policy, each developer needs to find the tools or solutions themselves.

The advantages of commercial software are the ease of use, no need for research and development, no need for responsibility and maintenance. But the disadvantages are:

a. Lack of flexibility, in case of the need for scaling in cost and system size, we can not remove some unnecessary function for cost reduction.

b. Vendor-Dependence: all platforms depending to the vendor, user must be use or change by vender's architecture e.g. software upgraded, hardware selection.

c. The maintenance period depends on vendor's policy.

From those reasons, to create the cost-effective system that readily for emergency use, the open source platform should be the solution.

In investigate the open source software, the first thing to be considered is the wide spread used. Since the tools are more available. The next issue is the stability, for the reliable reason. The ease of use is also important, therefore, the system should need the short period in development and easily to be learned by any user.

There are several Open Source platforms that support Voice over IP operation, such as Trixbox (for Linux CentOS), Asterisk(for Linux Ubuntu), AsteriskNOW (for Linux CentOS), FreeBSD etc. But the Linux OS that more readily for Server service is Cent OS. FreeBSD is suitable for the UNIX skilled level of user.

Asterisk

Asterisk is one IP-PBX application runs under Linux Ubuntu. But the stability is still weak (based on our experiment). The Graphic User Interface (GUI) application that support Asterisk is difficult in configuration and has some problem in real-time performance.

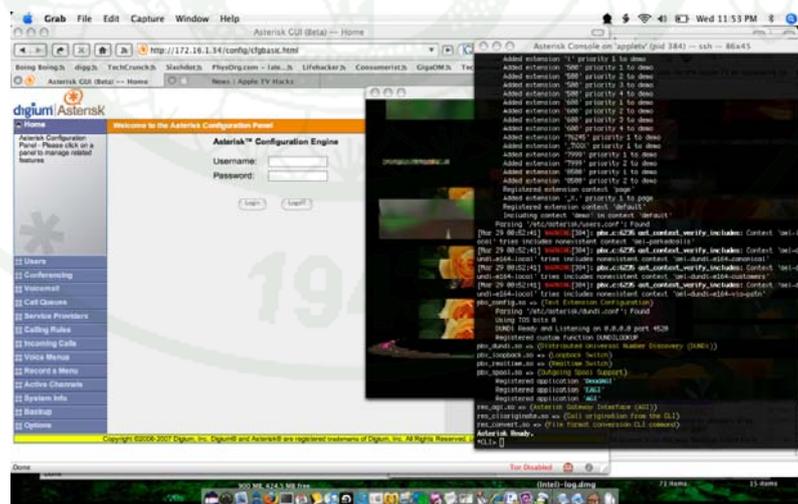


Figure 20 Asterisk Control Panel

Trixbox

Trixbox is an IP-PBX application runs under Linux Cent OS. The former name is Asterisk@home. Trixbox is developed from Asterisk-Based system by increase more user friendly GUI and Web-based Control Panel. From our Experiment, The disadvantage of Trixbox is the complexity of the control panel compared with Asterisk and AsteriskNOW.

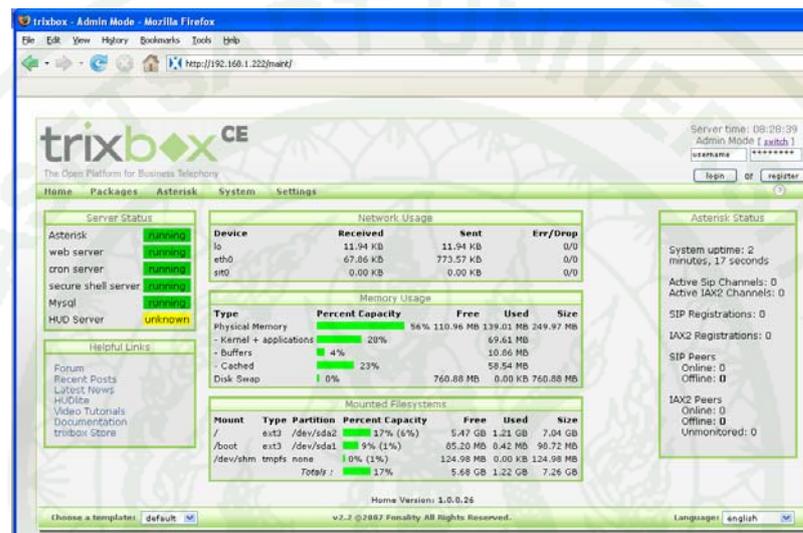


Figure 21 Trixbox Control Panel

Elastix

Elastix is an IP-PBX application runs under Linux Cent OS, developed from Asterisk-Based system, GUI and Web-based Control Panel added. But it is on the beginning period of development that still meets the unstable efficiency.

Askozia

Askozia is an IP-PBX application runs under FreeBSD system. The difference of ASkozia is ability in Removable Media installation. But in configuration, the user needs to have a knowledge base on FreePBX system.

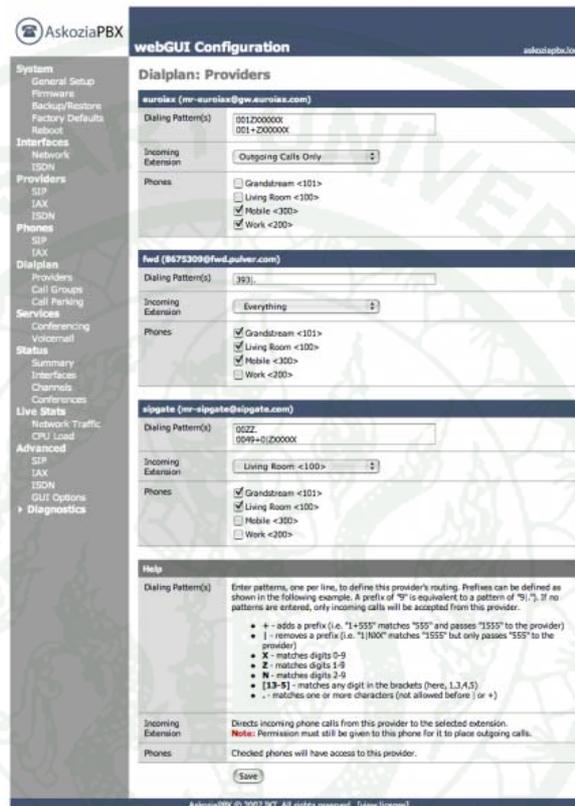


Figure 24 Askozia Control Panel

AsteriskWin32

AsteriskWin32 is an Asterisk-Based IP-PBX application runs under Windows system.



Figure 25 AsteriskWin32 Control Panel

AsteriskNOW

AsteriskNOW is an IP-PBX application runs under Linux Cent OS. The CentOS is more stable than other Linux platform such as Ubuntu, Debian, Slackware, PCLinuxOS or SUSE , for this reason, it's more suitable to perform as Server operation. AsteriskNOW is easy in use and configure. Moreover, the FreePBX Web-based GUI application that cooperated with AsteriskNOW has more user friendly, and it perform a real-time monitor and control.



Figure 26 FreePBX Control Panel

Table 5 Comparison between VoIP Solutions

Solution	Stability	Ease of Use	Supported CODECs	Notice
Asterisk	★★	★★	★★★★	Linux Ubuntu-Based
AsterkNOW	★★★★	★★★★	★★★★	Linux CentOS- Based
Elastix	★★★	★★★★	★★★★	Linux CentOS- Based
Trixbox	★★★★	★★★	★★★★	Linux CentOS- Based
Askozia	★★★	★★★	★★★★	FreeBSD - Based
Slast	★★★	★★	★★★★	Linux Slackware - Based
AsteriskWin32	★★★★	★★★	★★★	Windows-Based
3CX	★★★★	★★★	★★★	Windows-Based

★★★★ = Very Good

★★★ = Good

★★ = Fair

★ = Not Good

Based on our preliminary experiment results (shown in Table 5.), we can conclude that Asterisk Linux OS, Asterisk PBX, free codec, for system development and SIPp software and Wireshark analyzer, for system evaluation.

2. Develop necessary hardware

In hardware development for Radio over IP system, the performance of Radio over IP needs the Telephone-Radio controller circuit. The circuit including with 5 parts:

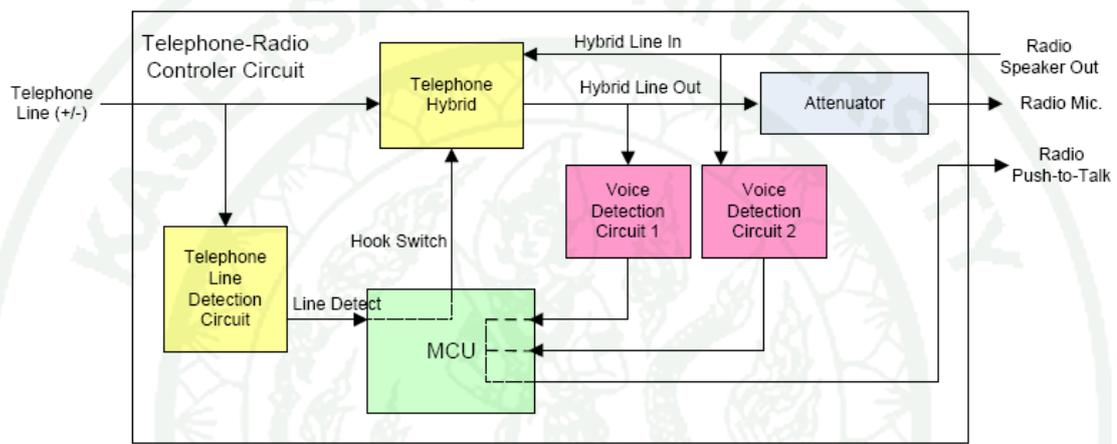


Figure 27 Block diagram of Telephone to Radio controller circuit

Essentially, we develop hardware for this system which is the VOX circuits and the Radio over IP gateway (Telephone Hybrid). This is important to our system because it makes a solution of Radio-Phone cooperation. By purchase, the VOX is the most expensive part of Radio over IP system. When developing by ourselves, we can reduced the cost and create custom function (threshold selection).

2.1 Telephone Hybrid

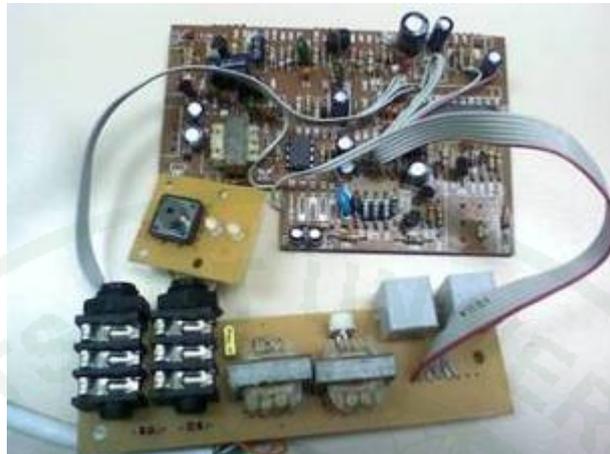


Figure 28 A Telephone Hybrid Circuit



Figure 29 A completed assembly of Telephone Hybrid Circuit

The circuit is designed to transform the normal telephone line and separate to Line in and Line out form. Therefore, the communication between telephones and radios can be enabled.

2.2 Telephone Line Detection

The circuit is designed to detect the telephone line based audio signal and transform to the MCU-readable signal. The MCU then can detect the Ring signal and switch the phone state to the off-hook or on-hook state.

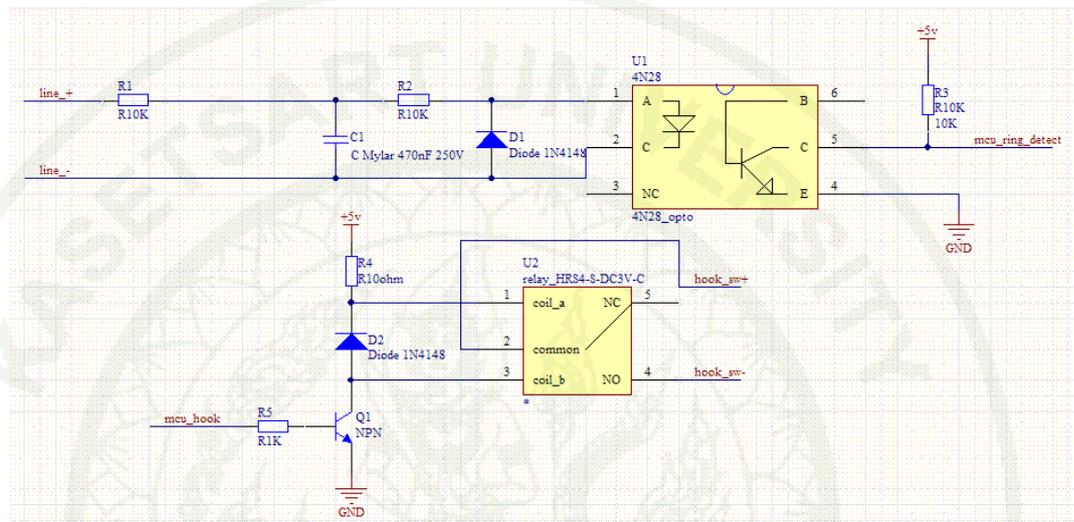


Figure 30 The diagram of Telephone Line Detection Circuit

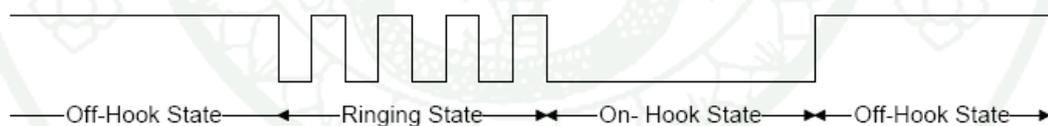


Figure 31 Example of telephone line diagram

The telephone line signal detection consists of R, C, Diode and Opto-Isolator.

The signals detected can be separated to 4 state:

2.2.1 Off-hooked State, where the signal level will be set to the Logic High.

2.2.2 Ringing State, the signal will be switched between Logic Low and Logic High with frequency 10-20 Hz.

2.2.3 On-hooked State, where the signal level will be set to the Logic Low.

2.3 Voice Detection

The circuit incorporates the necessary amplifiers, attenuators, level detectors, and control algorithm to form a high quality of VOX/PTT operation. It is the important part to control a Push-to-Talk Signal transmission.

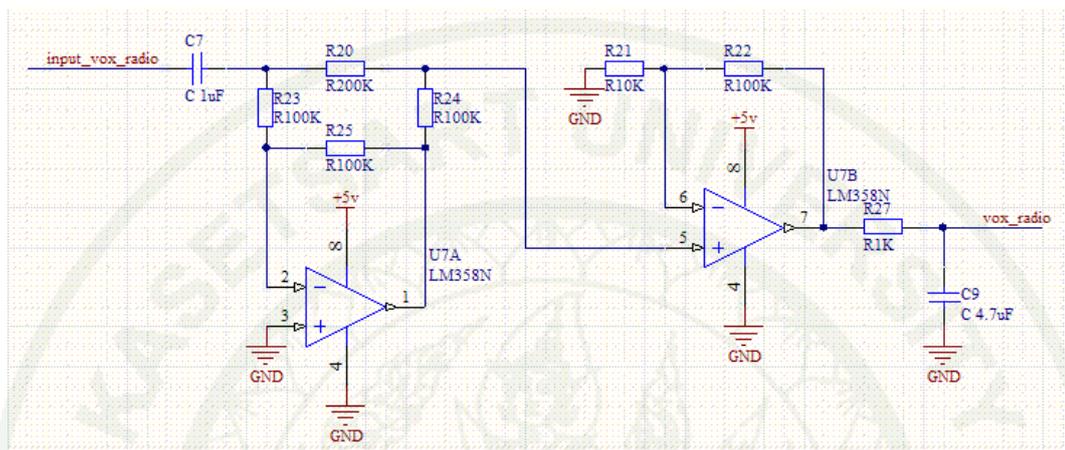


Figure 32 The diagram of Voice Detection Circuit

The Op Amp U7A and U7B transform the voice signal to single phase signal, and then filtered by RC filter (R27 and C9). The output signal can be approximated as a voice level.

2.4 Attenuator

This device is adopted to reduce the amplitude or power of a signal from the Hybrid telephone circuit where the Line Out signal is too high to feed forward to microphone.

In the circuit diagram, R is connected as PI model, and C is used for DC filtering.

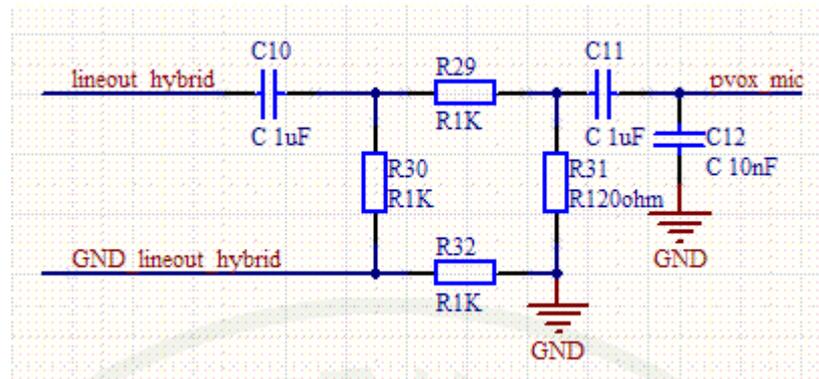


Figure 33 The diagram of Attenuator Circuit

2.5 MCU

This unit is the controller for all process including with signal detection, timer, and other cooperation. The MCU is connected to 2 necessary systems:

2.5.1 Telephone system: to detect the line state in order to hook and hang up control.

2.5.2 Radio system: to detect the voice level for a Push-To-Talk control.

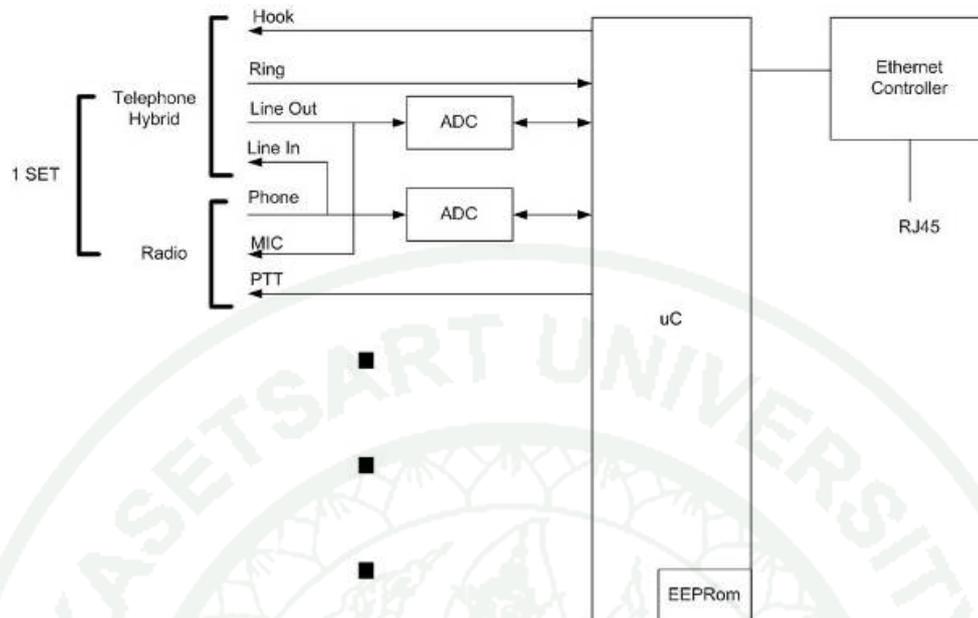


Figure 34 MCU Unit Diagram

2.6 Push to Talk

Push-to-Talk (PTT) is the property of a walkie-talkie mobile radio. It used for allow the speaker to talk via device to group members by pressing the PTT button. Since PTT is half duplex communication, only one group member is allowed to speak at a time, and the other members listen.

The traditional 2-way radio is not sustainable as convergence communication system technology. In RoIP communication, the system demands more than simple radio-to-radio communications. Some cellular telephones can allow PTT technologies. But, any alternative PTT communication supports the non-PTT telephone?

In this work, we apply an auto-answer mode to radio-radio or radio-phone cooperation to manage SIP-based push-to-talk services for Radio over IP application.

The purpose of the service is to incorporate the hybrid association of mobile radio devices, together with the telephones. With the implementation of Voice Operated eXchange (VOX) and Telephone Hybrid, telephone or any VoIP communication device can be accepted any incoming Push- to-talk sessions.

A PTT function allows transmission of the audio. Any time the user pushing the PTT button to speak on a hand held radio, voice will travel via radio waves to the other side end-user radio. The Carrier Operated Relay function (COR) is a receiver signal that gives a positive indication a carrier or signal is being received and the receiver is unscelched (Lin *et al.*, 2008). The device connected to the radio must have the ability to create the COR (Lin *et al.*, 2008). If the radio does not supply COR, the VOX (Voice Operated eXchange) canbe used to detect whenever the receiving radio voice signal exceeds a set threshold. The threshold can be varied for difference conditions.

In A/D and D/A conversion, the Analog Telephone Adaptor (ATA) devices are used, similar to the VoIP system. The difference is the RoIP system needs the Telephone to Radio Controller Circuit to support the Radio-Phone cooperation.

The controller circuit including with the MCU, Telephone Hybrid, Telephone Line Detection, Voice Detection, and Attenuator

The most significant parts that operate the PTT control scheme consist of the Telephone Hybrid and Voice detection.

A VOX (Voice Operated eXchange) function is implemented in the Voice Detection Circuit using the VAD (Voice Activity Detector) feature. When a voice signal is detected at the radio port, an internal Carrier Operated Squelch (COS) signal is generated and transmitted to the destination. This will also enable the transmission of voice packets over the IP network. A hang period is automatically applied to the VOX algorithm. Full duplex operation is supported. On the other hand, the circuit is also designed for the voice transmission protection in order to disable the voice transmission while the receiving signal is detected.

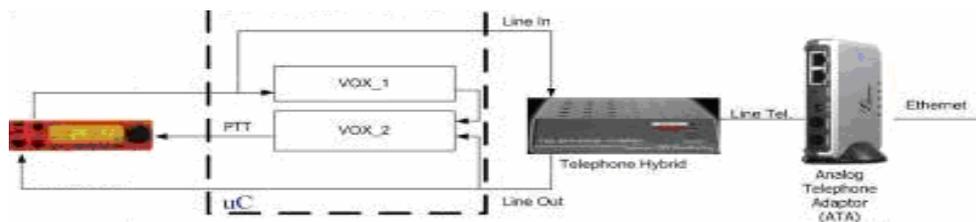


Figure 35 Radio-Phone cooperation via Telephone Hybrid circuit

Microphone

For the telephone user, the Telephone Line Detection will detect the hook state signal and the ringing signal from the phone line. Then the microcontroller will compare the Line Out signal with the threshold voltage. If the Line Out signal $>$ threshold, the VOX_2 will transmit the PTT signal to press the PTT. On the other hand, while the user from radio side is speaking, the VOX_1 will obstruct VOX_2 from transmitting signal to PTT by looping back the signal to the Line Out.

Speaker

We apply the SIP server related to our previous study which keeps compatible with SIP UAC, SIP UAS, and another SIP signaling. In pseudo datagram described below, the PTT signal is included.

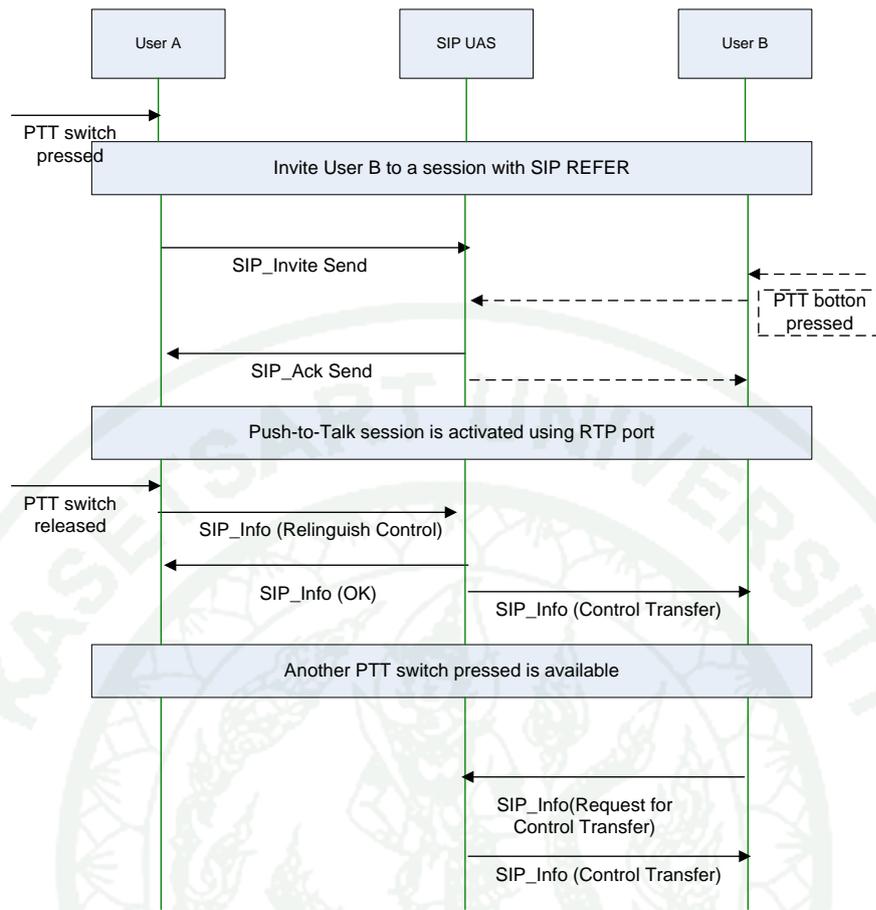


Figure 36 Signaling diagram of PTT control scheme

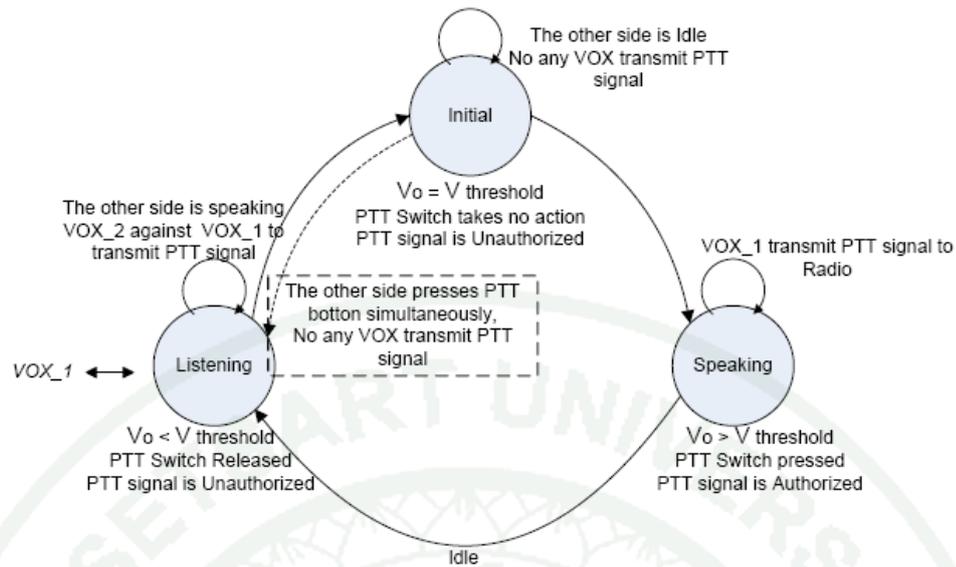


Figure 37 Single side of Push-to-Talk control scheme

State diagram in figure 32 describes the PTT performance over RoIP. For example, if one side (may be telephone) connected to VOX_1 and the other side (may be radio) connected to VOX_2, when the initial and speaking states are activated by the first side that connected to VOX_1, the VOX_1 will transmit the PTT signal to press the PTT, while looping back the signal to obstruct VOX_2 to transmit any signal. On the other hand, if the side that first reach the threshold is connected to VOX_2, the VOX_2 will transmit the PTT signal to press the PTT, as the same scheme but difference side. In case the both side press PTT simultaneously, no VOX can transmit any signal.

In threshold varying, we need to consider a surrounding noise, threshold may be varied upper or lower, and it certain effect to end-to end delay. The higher threshold would give more effectiveness to noise separation, but it also causes more delay. The lower delay makes the VOX sensitive to voice, and also sensitive ot noise.

From the experiment, where the PTT threshold was set to 5.0 volt, the end-to-end delays are shown in table 5.

Table 6 End-to-End Delay from Push-to-Talk

Device	PTT Method	End to End Delay [s]
Radio to Radio	Push PTT button key	2.0
Radio to Radio	VOX	1.0
Radio to Telephone	Push PTT button key	2.0
Radio to Telephone	VOX	1.0
Telephone to Radio	VOX	1.0

With 1.0 second of end-to-end delay, 5 V of threshold is acceptable in use for emergency communication where no clamor appeared.

3. Find a suitable configuration of the system

The appropriate configuration is obtained by the performance evaluation of the system. To find a guideline for suitable configuration under satellite broadband for emergency situation, the following performance tests were created.

a. The real environment test, under the 2/1 Mbps (The actual Downlink/Uplink bandwidth of IPstar broadband internet service.) The result from the simulation test under the limited bandwidth can lead us to consider the bandwidth management, e.g. codec allowance, video transmission allowance, to meet the possible quality of service guarantee.

b. The software simulation test. Since the limit number of communication devices. We can use the efficiency of software load test to extend the call load test by simulation. By this method we can determine the maximum number of call support or the system limitation in order to provide the call support in the future extend range of satellite broadband technology.

In evaluation of voice and video quality by using software monitoring, we can measure the end to end delay, jitter delay, and packet loss, and then find the appropriate compensation configuration such as de-jitter buffer, by empirical estimation.

3.1 System Architecture

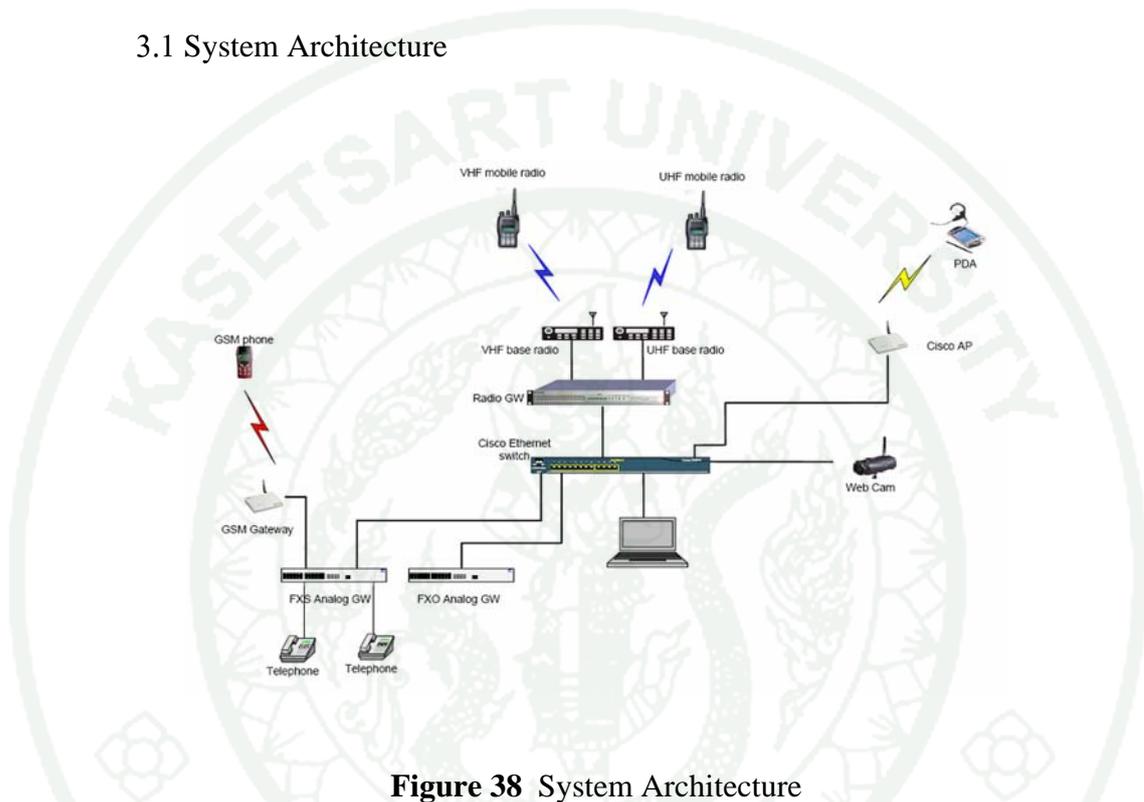


Figure 38 System Architecture

This IP-PBX system uses a single server for each node. The system organized by Linux Cent OS, handle with AsteriskNOW IP-PBX and open source applications, providing both voice and video applications support.

The prototype configuration for VoIP server section consisted of the following details: The Intel® Server SR1530SH ,Processor : 64-bit Multi-Core Intel® Xeon® 3200/3000 sequence 1.33 GHz, 8 GB RAM, and one Gigabit Ethernet Switch Cisco SRW224G4P.

Our system consists of the following:

1. Purchased Part Materials

Computer, VoIP Gateway, Network components, VoIP Phones, Telephones, Mobile Radios, Antennas, Softphones

2. Existing VoIP Software Solution

3. Developed hardware

System Configurations can be separated to two section; Asterisk Configuration and Hardware Configuration

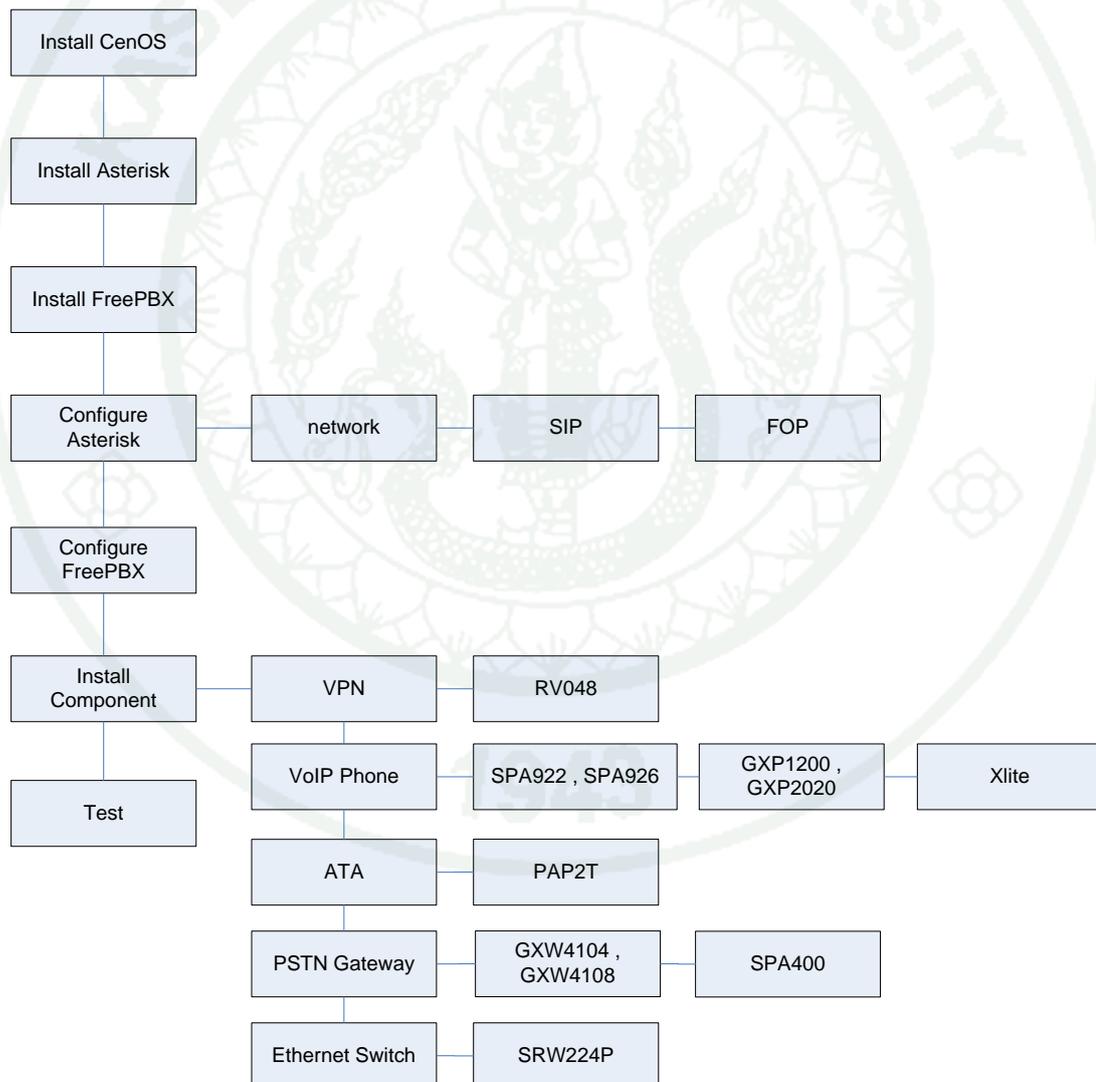


Figure 39 The diagram of Software Installation and System Configuration

3.1 Asterisk Configuration

3.1.1 Network Configuration

After Installation, first of all we have to assign the static IP address to the system. Since the SIP registration method needs a checking for IP address validity between SIP client devices and SIP server.

The important data are concern with network configuration. Here is the configuration detail for the system.

```
#nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
GNU nano 1.3.12 File: ifcfg-eth0
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0
BOOTPROTO=static
TYPE=Ethernet
USERCTL=no
IPV6INIT=no
PEERDNS=yes
IPADDR=192.168.55.100
NETMASK=255.255.0.0
HWADDR=00:0C:29:2E:78:47
ONBOOT=yes
```

3.1.2 CODEC allowance

The configuration for codec allowance is detailed below. To control the bandwidth consumption, the large payload CODEC should be disallowed for emergency communication via low bandwidth link.

```
.
#nano /etc/asterisk/sip_custom.conf
```

```
GNU nano 1.3.12 File: /etc/asterisk/sip_custom.conf Modified
allow=gsm
allow=h261
allow=h263
allow=h263a
allow=h264
videosupport=yes_

^G Get Help   ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify   ^W Where Is  ^U Next Page  ^U UnCut Text ^T To Spell
```

3.2 Flash Operator Panel Configuration

The connection between Graphic User Interface (GUI) pages and the Asterisk system is connected via Flash Operator Panel (FOP). Here we can configuration for real-time data appearance and system's control via http or other ports.

```

GNU nano 1.3.12      File: /etc/asterisk/manager.conf      Modified
;
; Asterisk Call Management support
;
[general]
enabled = yes
port = 5038
bindaddr = 0.0.0.0

[admin]
secret = amp111
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.0
read = system,call,log,verbose,command,agent,user,originate
write = system,call,log,verbose,command,agent,user,originate_

#include manager_additional.conf
#include manager_custom.conf

^G Get Help   ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify   ^W Where Is  ^U Next Page  ^U UnCut Text ^T To Spell

```

3.3 Dial plan Configuration

The dial plan is main concerns with SIP configuration (sip.conf), Extension configuration (extension.conf), channels configuration (chan_dahdi.conf) and Jitter Buffer Configuration.

3.3.1 SIP configuration

To setup the general service for the system such as account registration method, service authority, or service limitation. We can create or change all configurations.

To configure the appropriate values, we should consider the system's policy, such as the network connection, port allowance etc.

The lines below describe some example of SIP configuration.

```
#nano /etc/asterisk/sip.conf
```

```

[general]

#include sip_clients.conf
#include sip_register.conf
#include sip_trunks.conf

[general]
type=friend ; there are three levels -user, friend, peer
secret=password ; SIP Account password
qualify=yes ; Check for online state every 60 s
port=5060 ;Local SIP port
nat=yes|no

host=dynamic ; IP update enable for clients registration
setvar = variable = value ; setting Channel Variable for SIP Clients calling
dtmfmode=rfc2833
disallow=all ; always set disallow all before setting allowance
allow=g711
allow=g729
allow=alaw
allow=ulaw
context=from-internal
canreinvite=no
registertrying=yes
maxcallbitrate = <birtate> ; maximum bit rate allowed for video (kbps)
busylevel = <number> ; Maximum calls limited
call-limit = <number> ; limit number of call from this SIP accounts
rtptimeout = <seconds> ; period for disconnect when no RTP signal
timert1 = <milliseconds> ; Roundtrip Time

```

```

session-timers = <accept|originate|refuse> ; session-timer for "200 OK" signal
; between Asterisk(UAS) and Client(UAC)

```

3.3.2 Extension configuration

The extension file includes the setup the general service for the system such as account registration method, service authority, or service limitation. We can create or change all configurations.

To configure the appropriate values, we should consider the system's policy, such as the network connection, port allowance etc.

The lines below describe some example of SIP configuration.

```
#nano /etc/asterisk/extensions.conf
```

```

; Without loop context

[from-zaptel]                ; context for incoming from
                             ; outbound call
exten => s,1,Answer()        ; received call
exten => s,n,Dial(SIP/0101,15) ; Dial extension 0101 with
                             ; timeout = 15 s
exten => s,n,Hangup()

; With loop context (Needs Priority setting)

[from-zaptel]                ; context for incoming from
                             ; outbound call
exten => s,1,Answer()
exten => s,2,Set(COUNT=0)      ; Set counter = 0
exten => s,3,Background(th-mainmenu)
exten => s,4,Set(COUNT=${COUNT} + 1) ; increase counter

```

```

exten => s,5,GotoIf(${COUNT} = 3)?stopping:goingon) ; counter= 3 ?
                                                ; (for maximum loop )

```

```

exten => s,6(stopping),Hangup()           ; counter = 3 then
                                           ; Hangup()

exten => s,7(goingon),WaitExten()         ; counter < 3 go on continue

exten => t,1,Goto(s,3)
exten => i,1,Playback(th-invalid)
exten => i,2,Goto(s,3)                     ; invalid number and looping
                                           ; back to main menu

exten => s,1,Answer()                       ; received call
exten => s,2,Background(th-mainmenu)       ; play greeting message and menu
                                           ; selection

exten => s,3,WaitExten()                   ; wait for menu selection

exten => t,1,Goto(s,2)                     ; timeout (10s) and looping
                                           ; back to main menu

exten => i,1,Playback(th-invalid)          ; invalid number and looping
                                           ; back
                                           ; to main menu

[from-sip]                                ; context for inbound and outbound call

exten => 0101,1,Answer()
exten => 0101,n,Dial(SIP/0101,15)
exten => 0101,n,Hangup()

```

```

exten => 0102,1,Answer(0)
exten => 0102,n,Dial(SIP/0102,15)

```

```
exten => 0102,n,Hangup()
```

```
exten => _00XXXXXXXX,1,Dial(SIP/9/${EXTEN}) ; context for outbound call
```

```
exten => _668XXXXXXXX,1,Dial(SIP/9/${EXTEN})
```

```
exten => _662XXXXXXXX,1,Dial(SIP/9/${EXTEN})
```

3.3.3 Channels configuration

```
#nano /etc/asterisk/chan_dahdi.conf
```

```
#include dahdi-channels.conf
```

```
[channels]
```

```
context = from-pstn
```

```
language = en
```

```
callerid = asreceived
```

```
usecallerid = yes
```

```
threewaycalling = yes
```

```
transfer = yes
```

```
callwaiting = no
```

```
echocancel = yes
```

```
rxgain = 0.0
```

```
txgain = 0.0
```

```
transfer = yes
```

```
canpark = yes
```

```
cancallforward = yes
```

```
callreturn = yes
```

```
immediate = no
```

```
progzone = us
```

```
tonezone = 0
```

```
faxdetect = yes
```

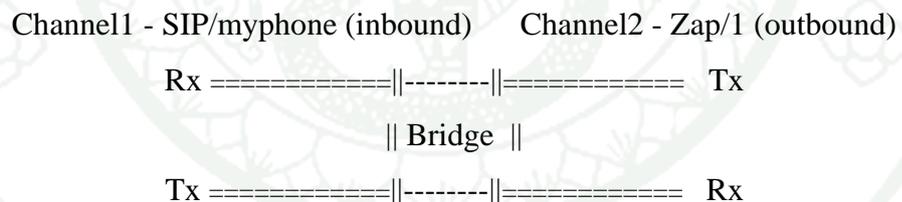
```

callprogress = yes
ringtimeout=8000
jbenable = yes
busydetect = yes
busycount = 5
pattern=500,500
echotraining = yes

```

3.3.4 Jitter Buffer Configuration

One of the major advantages of IAX2 over SIP is the IAX2 jitterbuffer (in Asterisk 1.2.x and higher). A simple scenario below represent where a SIP phone makes a call out via Zap PSTN channel. The Rx represents where audio is received from the network and comes into Asterisk. The Tx represents where audio is taken from Asterisk and transmitted back out over the network.



(Source: <http://www.asterisk.org/node/48317>)

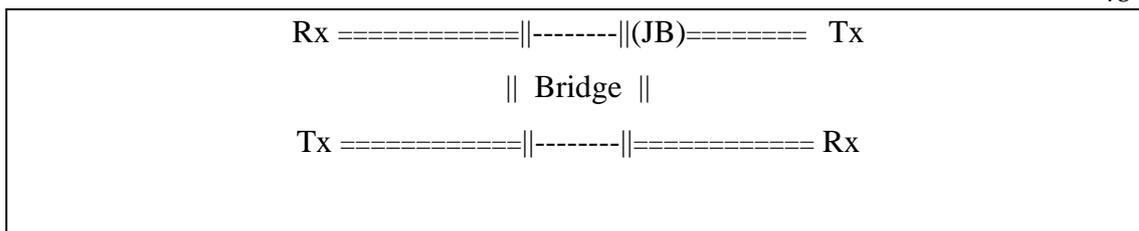
In this case, the audio received on the SIP channel is coming in over the network and needs to be de-jittered. Furthermore, the bridge sees that the channel on the other side is not an IP channel, so it actually makes sense to use a jitterbuffer.

Here is where the bridge puts the jitterbuffer: (noted using JB)

```

Channel1 - SIP/myphone (inbound)   Channel2 - Zap/1 (outbound)

```



(Source: <http://www.asterisk.org/node/48317>)

So, in this illustration, the jitterbuffer is sitting in the path of audio coming from the SIP channel, but before it goes out to the Zap channel. However, it actually lives on the Zap channel. Because of this, the configuration for this is actually done in `zapata.conf` (in some version of Asterisk, but for this version we can configure at `dahdi-channels.conf` or `sip_general_custom.conf`). In order to avoid the complex on each part of configuration, we will use the `sip_general_custom.conf` for jitterbuffer configuration.

Here is some example:

#nano /etc/asterisk/sip_general_custom.conf

```
;Enable Jitter Settings
jitterbuffers=4
jbenable=yes
jbforce=no
jbimpl=fixed
;jbimpl=adaptive
jbmaxsize=500
jbresyncthreshold=100
jblog=yes
;tos=0x18
```

;Enable Jitter Settings

jitterbuffers=4

<code>jbenable=yes</code>	
<code>jbforce=no</code>	<i>; Forces the use of a jitterbuffer on the receive side</i>
<code>jbimpl=fixed</code>	<i>; Jitterbuffer implementation, used on the receiving side</i>
<code>;jbimpl=adaptive</code>	
<code>jbmaxsize=500</code>	<i>; Max length of the jitterbuffer in milliseconds.</i>
<code>jbresyncthreshold=100</code>	<i>; Jump in the frame timestamps over which the jitterbuffer is resynchronized. Useful to improve the quality of the voice, with big jumps in/broken timestamps, usually sent from exotic devices and programs. Defaults to 1000.</i>
<code>jblog=yes</code>	<i>; Enables jitterbuffer frame logging.</i>
<code>;tos=0x18</code>	<i>; the tos bits are set individually and in combination</i>

At the log file we can see the details of jitterbuffer change like:

```

JB_GET {now=59259}: Delivered frame with ts=397012 and len=20
JB_GET {now=59264}: now < next=59279
JB_PUT {now=59279}: Queued frame with ts=397232 and len=20
JB_GET {now=59279}: Delivered frame with ts=397032 and len=20
JB_GET {now=59285}: now < next=59299
JB_PUT {now=59297}: Queued frame with ts=397252 and len=20
JB_GET {now=59297}: now < next=59299
JB_GET {now=59298}: now < next=59299
JB_GET {now=59299}: Delivered frame with ts=397052 and len=20
JB_GET {now=59304}: now < next=59319

```

3.4 Network Configuration

The external connection between RoIP networks can be deployed to the private internet network concepts. In the real situation, only the authorized person can access

to the network for security policy. Therefore, the concept of private network is implemented to our system. Some useful methods are VPN: Virtual Private Network and DDNS: Dynamic Domain Name Server. Both can support the best performance to the system.

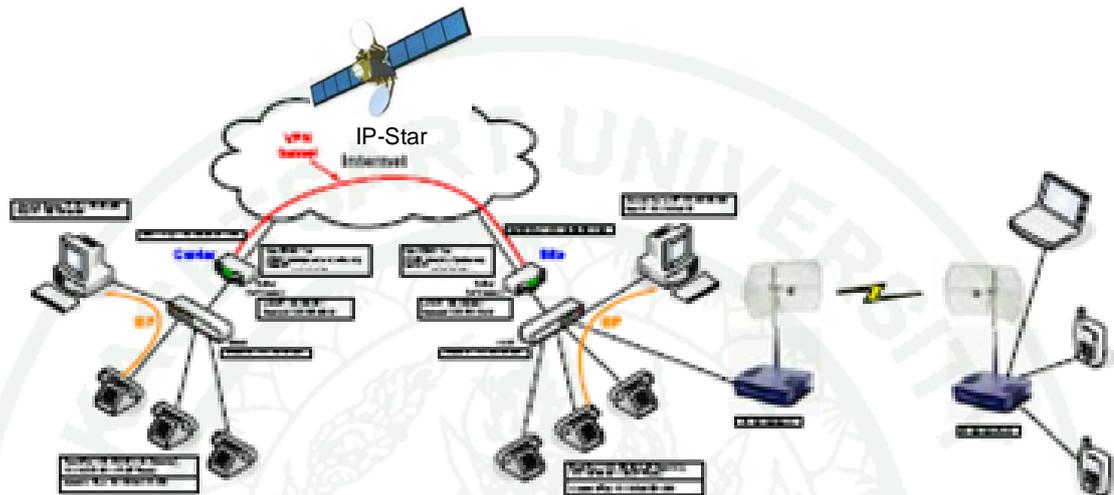


Figure 40 The diagram of network connection via satellite

By these applications, the emergency communication system can be completely set and perform the mission via broadband satellite within half an hour. The quality of communication between external links or network to network is depend on the quality of satellite link. The quality of internal network communication is depends on the physical features of the topography, and it is more controllable.

By Asterisk interface, the trunk communication of each base can implement both SIP Trunk and IAX Trunk.

SIP trunk is a service offered by an ITSP (Internet Telephony Service Provider) to use SIP to set up communications between an enterprise PBX and the ITSP. The basic trunking system contains one channel that serves as a control or "handshake" channel, known as a data channel.

IAX (Inter-Asterisk eXchange): When trunking with IAX2, only the used bandwidth is allocated at all times. Other TDMoIP (Time Division Multiplexing-over-

IP) protocols used for trunking always allocate a certain amount of bandwidth to keep all channels open. IAX trunking allows multiple voice streams to share a single "trunk" to another server, reducing overhead created by IP packets.

4. Performance Analysis

After Configurations, the system needs the functions evaluation test, such as the Web-based GUI application, the function of call setup, conference setup, voice recording etc. to observed the performance and rechecked the configuration, and then went to the field test and software simulation test.



Figure 41 The Web-based GUI



Figure 42 Set up a call and video transmission to evaluate the call functions and analyze the Voice quality, Delay, Jitter and Bandwidth used.

4.1 Field Test

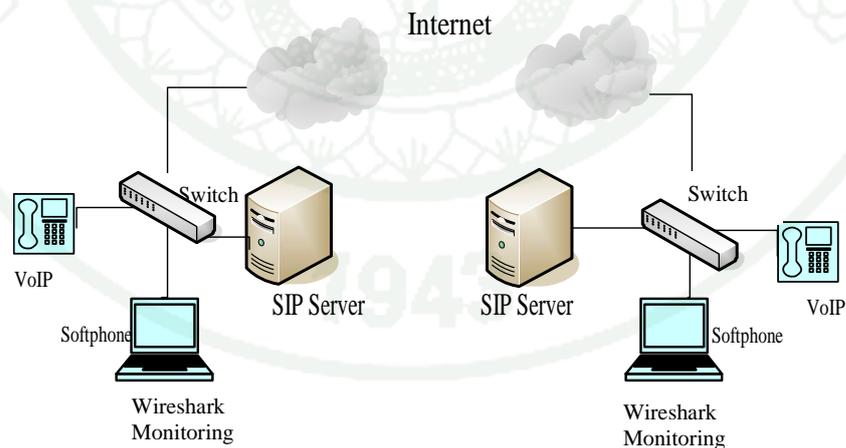


Figure 43 Performance Test of voice and video transmission via IPStar Broadband Internet

The broadband satellite link via IPstar has been test for the distance of 900 km between network to network (from Chiang Mai province in the north of Thailand to Bangkok).



Figure 44 The field test of the RoIP interoperability system interconnection between two network bases via satellite

To monitoring the voice and video transmission via satellite link, we set the performance test via IPStar Broadband, 2Mbps, no VAD or silence suppression. The equipment and devices are:

- 4.1.1 Two Server PC
- 4.1.2 Two Switches
- 4.1.3 Two VoIP Phone, for voice communication
- 4.1.4 Two client PC and Softphone, for video communication

In this experiment, the codec used for voice transmission are G.711 (bitrate 64 kbps) (G.711 Recommendation, 2004) and G.729 (bitrate 8 kbps) (G.729 Recommendation, 2007) . For video transmission, The bitrate supported for H.264 CODEC according to RFC 3550 (ITU-T H.264 SERIES H, 2005) RTP payload format depends on the sampling rate and number of channels selected. Each manufacturer customizes its H.264 profile to suit its needs, which means no two H.264 profiles are necessarily created equal ([www.iqeye.com/iqeye/H.264_Considerations .pdf](http://www.iqeye.com/iqeye/H.264_Considerations.pdf), 2009).

For the worst case scenarios we used variable bit rate (VBR) where the bandwidth will increase dramatically to attempt to achieve this purpose.

Table 7 G.711 and G.729 CODEC Information

Codec Information			
CODEC & Bit Rate (Kbps)	CODEC Sample Size (Bytes)	CODEC Sample Interval (ms)	Mean Opinion Score (MOS)
G.711 (64 Kbps)	80 Bytes	10 ms	4.1
G.729 (8 Kbps)	10 Bytes	10 ms	3.92

Table 8 Bandwidth Calculations for each CODEC

Bandwidth Calculations					
Voice Payload Size (Bytes)	Voice Payload Size (ms)	Packets Per Second (PPS)	Bandwidth MP or FRF.12 (Kbps)	Bandwidth w/cRTP MP or FRF.12 (Kbps)	Bandwidth Ethernet (Kbps)
G.711-160 Bytes	20 ms	50	82.8 Kbps	67.6 Kbps	87.2 Kbps
G.729-20 Bytes	20 ms	50	26.8 Kbps	11.6 Kbps	31.2 Kbps

Bitrate Calculation

G.711: 160 bytes * 8 bits/bytes * 50 packets / second = 64,000 bits per second
 G.729: 20 bytes * 8 bits/bytes * 50 packets / second = 8,000 bit per second

Bandwidth Calculations per 1 call, by approximately:

CODEC g711a

197 bytes (headers + payload) / 160 bytes payload only * 64,000 bit per second =
78,800 bits per second

CODEC g729a

57 bytes (headers + payload) / 20 bytes payload only * 8,000 bits per second = 22,800
bits per second

There are some difference between transmission via SIP Trunk and IAX2 Trunk. The bandwidth uses by IAX (Inter-Asterisk eXchange) is less than the one uses by SIP since the messages are binary instead of text messages (SIP), but it does not support RTP protocol, since IAX uses only one port for everything.

Bandwidth Calculations per 1 call, by Software:

Incoming Bandwidth	Outgoing Bandwidth
Calls: 1	Calls: 1
RTP: 4.69 Kbps	RTP: 4.69 Kbps
UDP: 3.13 Kbps	UDP: 3.13 Kbps
IP: 7.81 Kibps	IP: 7.81 Kibps
Protocol: SIP	Protocol: SIP
Audio Codec: 64.00g.711 Kbps	Audio Codec: 64.00g.711 Kbps
*SIP overhead is disregarded!	*SIP overhead is disregarded!
Incoming bandwidth:	Outgoing bandwidth:
83.61 Kbps	83.61 Kibps
0.08 Mbps	0.08 Mbps
10.45 KBps	10.45 KBps
0.01 MBps	0.01 MBps
Total bandwidth (incoming and outgoing):	
167.22 Kbps	
0.16 Mbps	
20.9 KBps	
0.02 MBps	

Figure 45 Bandwidth Calculations per 1 call using SIP Trunk, G.711 codec

Incoming Bandwidth		Outgoing Bandwidth	
Calls: 1		Calls: 1	
RTP: 4.69 Kbps		RTP: 4.69 Kbps	
UDP: 3.13 Kbps		UDP: 3.13 Kbps	
IP: 7.81 Kibps		IP: 7.81 Kibps	
Protocol: IAX2 REGULAR		Protocol: IAX2 REGULAR	
Audio Codec: 64.00g.711 Kbps		Audio Codec: 64.00g.711 Kbps	
*IAX2 REGULAR is not using RTP or RTCP!		*IAX2 REGULAR is not using RTP or RTCP!	
Incoming bandwidth:	76.5 Kbps	Outgoing bandwidth:	76.5 Kibps
	0.07 Mbps		0.07 Mbps
	9.56 KBps		9.56 KBps
	0.01 MBps		0.01 MBps
Total bandwidth (incoming and outgoing):		153 Kbps	
		0.15 Mbps	
		19.13 KBps	
		0.02 MBps	

Figure 46 Bandwidth Calculations per 1 call using IAX2 Trunk, G.711 codec

Incoming Bandwidth		Outgoing Bandwidth	
Calls: 1		Calls: 1	
RTP: 4.69 Kbps		RTP: 4.69 Kbps	
UDP: 3.13 Kbps		UDP: 3.13 Kbps	
IP: 7.81 Kibps		IP: 7.81 Kibps	
Protocol: SIP		Protocol: SIP	
Audio Codec: 8.00g.729a Kbps		Audio Codec: 8.00g.729a Kbps	
*SIP overhead is disregarded!		*SIP overhead is disregarded!	
Incoming bandwidth:	24.81 Kbps	Outgoing bandwidth:	24.81 Kibps
	0.02 Mbps		0.02 Mbps
	3.1 KBps		3.1 KBps
	0 MBps		0 MBps
Total bandwidth (incoming and outgoing):		49.62 Kbps	
		0.05 Mbps	
		6.2 KBps	
		0.01 MBps	

Figure 47 Bandwidth Calculations per 1 call using SIP Trunk, G.729codec

Incoming Bandwidth		Outgoing Bandwidth	
Calls: 1		Calls: 1	
RTP: 4.69 Kbps		RTP: 4.69 Kbps	
UDP: 3.13 Kbps		UDP: 3.13 Kbps	
IP: 7.81 Kibps		IP: 7.81 Kibps	
Protocol: IAX2 REGULAR		Protocol: IAX2 REGULAR	
Audio Codec: 8.00g.729a Kbps		Audio Codec: 8.00g.729a Kbps	
*IAX2 REGULAR is not using RTP or RTCP!		*IAX2 REGULAR is not using RTP or RTCP!	
Incoming bandwidth:	20.5 Kbps	Outgoing bandwidth:	20.5 Kibps
	0.02 Mbps		0.02 Mbps
	2.56 KBps		2.56 KBps
	0 MBps		0 MBps
Total bandwidth (incoming and outgoing):		41 Kbps	
		0.04 Mbps	
		5.13 KBps	
		0.01 MBps	

Figure 48 Bandwidth Calculations per 1 call using IAX2 Trunk, G.729 codec

Incoming Bandwidth		Outgoing Bandwidth	
Calls: 1		Calls: 1	
RTP: 4.69 Kbps		RTP: 4.69 Kbps	
UDP: 3.13 Kbps		UDP: 3.13 Kbps	
IP: 7.81 Kibps		IP: 7.81 Kibps	
Protocol: SIP		Protocol: SIP	
Audio Codec: 64.00g.711 Kbps		Audio Codec: 8.00g.729a Kbps	
*SIP overhead is disregarded!		*SIP overhead is disregarded!	
Incoming bandwidth:	83.61 Kbps	Outgoing bandwidth:	24.81 Kibps
	0.08 Mbps		0.02 Mbps
	10.45 KBps		3.1 KBps
	0.01 MBps		0 MBps
Total bandwidth (incoming and outgoing):		108.42 Kbps	
		0.11 Mbps	
		13.55 KBps	
		0.01 MBps	

Figure 49 Bandwidth Calculations per 1 call using SIP Trunk, G.711 and G.729 transcoder

Incoming Bandwidth		Outgoing Bandwidth	
Calls: 1		Calls: 1	
RTP: 4.69 Kbps		RTP: 4.69 Kbps	
UDP: 3.13 Kbps		UDP: 3.13 Kbps	
IP: 7.81 Kibps		IP: 7.81 Kibps	
Protocol: IAX2 REGULAR		Protocol: IAX2 REGULAR	
Audio Codec: 64.00g.711 Kbps		Audio Codec: 8.00g.729a Kbps	
*IAX2 REGULAR is not using RTP or RTCP!		*IAX2 REGULAR is not using RTP or RTCP!	
Incoming bandwidth:	76.5 Kbps	Outgoing bandwidth:	20.5 Kibps
	0.07 Mbps		0.02 Mbps
	9.56 KBps		2.56 KBps
	0.01 MBps		0 MBps
Total bandwidth (incoming and outgoing):		97 Kbps	
		0.09 Mbps	
		12.13 KBps	
		0.01 MBps	

Figure 50 Bandwidth Calculations per 1 call using IAX2 Trunk, G.711 and G.729 transcoder

4.2 Software Simulation Test

In the previous section, we perform experiments based on bandwidth limited satellite link to evaluate system performance under limited bandwidth scenarios. In this section, we try to evaluate the VoIP software performance under virtually unlimited network bandwidth (100Mbps Ethernet).

The performance test was set to monitoring the characteristics of the system on both service quality and voice quality.

Service quality: can be determined by the reliability on traffic management such as the call setup time, the call setup failure, disconnect probability etc.

Voice quality: can be analyzed from the transport signals such as SIP messages timestamps, latency, jitter and packet loss.

The important issue is a relationship between a service efficiency and a voice quality.

4.2.1 Test Methodology

The test used the SIPp 3.1, a free open software test tool / traffic generator for the SIP protocol. The SIPp is licensed under the GNU General Public License. It includes a few basic of Benchmarking SIP Server performance using for emulating user agents calling a SIP server (Razvan Rughinis and Cristian Iconaru, 2009). Reaching high call rates and/or high number of simultaneous SIP calls is possible.

4.2.2 SIP calls monitoring

The traffic load is distributed across the SIP server using Cisco SRW224G4P Gigabit Switch. The SIPp is installed on a client computer. The characteristic of average response time and call retransmissions can be monitored by the SIP client computer.

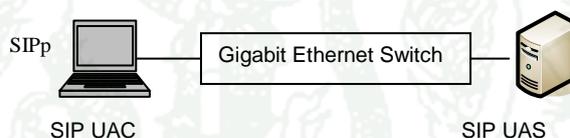


Figure 51 Experiment for system's response

4.2.3 SIP Traffic Model

The communications between SIP client and SIP server use both UDP protocol. And a stream of the Real-Time Transport Protocol (RTP) packets has been generated and transported to the traffic simulations.

RTP runs on top of transport protocols, typically UDP, and provides real-time applications with end-to-end delivery services such as payload type identification and delivery monitoring. RTP provides transport of data contains with a

message sequence number to allow applications to detect packet loss, packet duplication, or packet reordering.

By using an XML pattern for a SIP Call Flow, both voice and video traffic signals were generated. To set up an voice transmission session, The simulation was assigned to use G.711 CODEC. And in video transmission, H.264 video supported CODEC was used. G.711 - is the widely used PCM standard, encoding 8 bit speech samples at 8000 samples per second, giving 64 kilobits per second of digital speech data.

Some characteristics of G.711 codec are shown in Table 9.

4.2.4 Protocols for Simulation

Protocols used for software simulation

G.711 – Its formal name is Pulse code modulation (PCM) of voice frequencies. It is an ITU-T standard for audio companding with 64 kbps bit rate.

Table 9 Some Characteristics of G.711 CODEC

Standard	Bandwidth	Sample period	Ethernet Bandwidth
ITU-T G.711	64 kbps	20 ms	95.2 kbps

DTMF rfc 2833 – (DTMF = Dual Tone Multi Frequency) uses two tones to represent each key on the touch pad and then transform to 16 digits 1,2,.....8,9,0,*,#,a,b,c,d. DTMF is used in PSTN telephone.

H.264 - The bitrate supported for H.264 CODEC according to RFC 3550 (ITU-T H.264 SERIES H, 2005) RTP payload format depends on the sampling rate and number of channels selected. Each manufacturer customizes its H.264 profile to suit its needs, which means no two H.264 profiles are necessarily created equal

(H.264 Considerations, 2005). For the worst case scenarios we used variable bit rate (VBR) where the bandwidth will increase dramatically to attempt to achieve this purpose.

RTP – (RTP = Real Time Protocol) RTP provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. It does not address resource reservation and does not guarantee quality-of-service for real-time services.

4.2.5 Configuration at SIP server

4.2.5.1 Configuration at sip.conf

```
tos = none
dtmfmode = info
allow = alaw
allow = ulaw

[127.0.0.1]
type = friend
callerid = 112
host = dynamic
canreinvite = no
context = from-sipp
dtmfmode = rfc2833
quality=2000
```

Description:

- The command: tos (Type Of Service) means QoS managements
- The command allow = alaw, allow = ulaw: means CODEC allowance for G 711a or G 711u
- The command context = from-sipp: means using context named “from-sipp” in call controls from answer to hang up procedures.

4.2.5.2 Configuration at extensions.conf

```

[general]
Static = yes
Writeprotect = yes
Autofallthrough = yes
Clearglobalvars = yes
Priorityjumping = yes

[globals]

[from-sipp]
Exten => _x.,1,Answer
Exten => _x.,2,Dial(Zap/g1/111)
Exten => _x.,3,HangUp

[from-zaptel]
Exten => _x.,1,Answer
Exten => _x.,2,playback(demo-congrats)
Exten => _x.,3,HangUp

```

Description:

- At context “from-sipp”, the priority set are plan as :

1, Answer

2, Dial (Zap/g1/111); transfer the call to Zap/g1/111 (FXO channel)

that use the context named from- zaptel (Asterisk ZAP channels, 2010)

- At context “from-zaptel”, playback (demo-congrats) means transfer the call to IVR for the duration time assigned.

3, HangUp

4.2.5.3 .xml scripts for signal generating

1. The voice signal generating programming has some important part like.

```
<scenario name="UAC with media">
```

```
.....
```

```

INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0.
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch].
From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_num]
To: sut <sip:[service]@[remote_ip]:[remote_port]>.
Call-ID: [call_id].
CSeq: 1 INVITE.
Contact: sip:sipp@[local_ip]:[local_port].
Max-Forwards: 70.
Subject: Performance Test.
Content-Type: application/sdp.
Content-Length: [len].

v=0.
o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip].
s=-.
c=IN IP[local_ip_type] [local_ip].
t=0 0.
m=audio [auto_media_port] RTP/AVP 8 101.
a=rtpmap:8 PCMA/8000.
a=rtpmap:101 telephone-event/8000.
a=fmtp:101 0-11,16.
.
<!-- Play a pre-recorded PCAP file (RTP stream) -->.
<nop>.
  <action>.
    <exec play_pcap_audio="pcap/g711a.pcap"/>.
  </action>.
</nop>.
.
<!-- Pause 8 seconds, which is approximately the duration of the -->.
<!-- PCAP file -->.
<pause milliseconds="8000"/>.
.
<!-- Play an out of band DTMF '1' -->.
<nop>.
  <action>.
    <exec play_pcap_audio="pcap/dtmf_2833_1.pcap"/>.
  </action>.
</nop>.
.
<pause milliseconds="1000"/>.
.
<!-- The 'crlf' option inserts a blank line in the statistics report. -->.
<send retrans="500">.

```

Description:

- The command `<exec play_pcap_audio="pcap/g711a.pcap"/>` means voice signal generating using Codec G 711a

- The command `<exec play_pcap_audio="pcap/dtmf_2833_1.pcap"/>` means voice signal generating using Codec DTMF rfc 2833

2. The video signal generating programming has some important part like.

```

</send>.
<!-- Play a pre-recorded PCAP file (RTP stream) -->.
<nop>.
  <action>.
    <exec play_pcap_audio="pcap/g711a.pcap"/>.
  </action>.
</nop>.

<nop>.
  <action>.
    <exec play_pcap_video="MMdemoMainmenuH263.pcap"/>.
  </action>.
</nop>.

<!-- Pause 8 seconds, which is approximately the duration of the -->.
<!-- PCAP file -->.
<pause milliseconds="8000"/>.

<!-- Play an out of band DTMF '1' -->.
<nop>.
  <action>.
    <exec play_pcap_audio="pcap/dtmf_2833_1.pcap"/>.
  </action>.
</nop>.

<pause milliseconds="1000"/>.

<!-- The 'crLf' option inserts a blank line in the statistics report. -->.
<send retrans="500">.

```

Description:

- The command `<exec play_pcap_audio="pcap/g711a.pcap"/>` means voice signal generating using CODEC G 711a

- The command `<exec play_pcap_video="MMdemoMainmenuH263.pcap"/>` means video signal generating using CODEC H263

- The command `<exec play_pcap_audio="pcap/dtmf_2833_1.pcap"/>` means voice signal generating using CODEC DTMF rfc 2833

SIPp Command

```
>sipp -sf file name -r round -rp round time(in ms) -d time duration -l call limit IP
address of SIP server
>sipp -sf videouac.xml -r 10 -rp 1000 -d 10000 -l 10 192.168.55.100
```

Description:

Generate the 10 calls (-r 10) in 1 second(-rp 1000 ms), the call duration is 10 seconds (-d 10000 ms). The live calls hold in the system are not exceed 10 calls (-l 10).

The log file below shows the system's response during SIPp simulation. From this example, in order to observable, the system was forced to generate voice transmission at 1 call per 1 second every 5 second, the time duration of each call is 10 second.

```
>sipp -sf videouac.xml -r 1 -rp 5000 -d 10000 -l 1 192.168.55.100
```

Description:

- The command `<exec play_pcap_audio="pcap/g711a.pcap"/>` means voice signal generating using CODEC G 711a

- The command `<exec play_pcap_video="MMdemo Mainmenu H263 .pcap"/>` means video signal generating using CODEC H263

- The command `<exec play_pcap_audio="pcap/dtmf_2833_1.pcap"/>` means voice signal generating using CODEC DTMF rfc 2833

SIPp Command

```
>sipp -sf file name -r round -rp round time(in ms) -d time duration -l call limit IP
address of SIP server
>sipp -sf videouac.xml -r 10 -rp 1000 -d 10000 -l 10 192.168.55.100
```

Description:

Generate the 10 calls (-r 10) in 1 second(-rp 1000 ms), the call duration is 10 seconds (-d 10000 ms). The live calls hold in the system are not exceed 10 calls (-l 10).

The log file below shows the system's response during SIPp simulation. From this example, in order to observable, the system was forced to generate voice transmission at 1 call per 1 second every 5 second, the time duration of each call is 10 second.

```
>sipp -sf videouac.xml -r 1 -rp 5000 -d 10000 -l 1 192.168.55.100
```

```

-- Executing [service@from-sip-external:1] NoOp("SIP/192.168.55.230:5060-04ba7008", "Received an RTP packet") in new stack
-- Executing [service@from-sip-external:2] Set("SIP/192.168.55.230:5060-04ba7008", "DID=service") in new stack
-- Executing [service@from-sip-external:3] Goto("SIP/192.168.55.230:5060-04ba7008", "s,1") in new stack
-- Goto (from-sip-external,s,1)
-- Executing [s@from-sip-external:1] GotoIf("SIP/192.168.55.230:5060-04ba7008", "1?from-trunk") in new stack
-- Goto (from-trunk,service,1)
-- Executing [service@from-trunk:1] Set("SIP/192.168.55.230:5060-04ba7008", "__FROM_DID=service") in new stack
-- Executing [service@from-trunk:2] NoOp("SIP/192.168.55.230:5060-04ba7008", "Received an RTP packet") in new stack
-- Executing [service@from-trunk:3] Goto("SIP/192.168.55.230:5060-04ba7008", "s,a2") in new stack
-- Goto (from-trunk,s,2)
-- Executing [s@from-trunk:2] Answer("SIP/192.168.55.230:5060-04ba7008", "") in new stack
-- Executing [s@from-trunk:3] Wait("SIP/192.168.55.230:5060-04ba7008", "2") in new stack
-- Executing [s@from-trunk:4] Playback("SIP/192.168.55.230:5060-04ba7008", "ss-noservice") in new stack
-- <SIP/192.168.55.230:5060-04ba7008> Playing 'ss-noservice.gsm' (language 'en')
-- Executing [s@from-trunk:5] SayAlpha("SIP/192.168.55.230:5060-04ba7008", "service") in new stack
-- <SIP/192.168.55.230:5060-04ba7008> Playing 'letters/s.gsm' (language 'en')
-- <SIP/192.168.55.230:5060-04ba7008> Playing 'letters/e.gsm' (language 'en')
-- <SIP/192.168.55.230:5060-04ba7008> Playing 'letters/r.gsm' (language 'en')
-- <SIP/192.168.55.230:5060-04ba7008> Playing 'letters/v.gsm' (language 'en')
= Spawn extension (from-trunk, s, 5) exited non-zero on 'SIP/192.168.55.230:5060-04ba7008'
-- Executing [h@from-trunk:1] Hangup("SIP/192.168.55.230:5060-04ba7008", "") in new stack
= Spawn extension (from-trunk, h, 1) exited non-zero on 'SIP/192.168.55.230:5060-04ba7008'

```

Wireshark

The transmission/receiving packets from each SIP call includes payload type identification, sequence numbering, and timestamping can be monitored and recorded by Wireshark software tool Version 0.99.7 (SVN Rev 23910).

Wireshark is a network protocol analyzer for Unix and Windows, it available under the GNU General Public License. With packet filtering, we considered only the SIP RTP protocol packets.

The screenshot displays the Wireshark interface with the following sections highlighted:

- Section Menu & Command:** Shows the menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Help) and the toolbar.
- Section Capture Packets:** A table listing captured packets with columns for No., Time, Source, Destination, Protocol, and Info.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.1	224.0.0.1		
2	0.875019	192.168.1.100	239.255.255.250		
3	3.375134	192.168.1.100	224.0.0.251	IGMP	V2 Membership Report
4	16.687710	Intel_46:cf:0d	Broadcast	ARP	who has 192.168.1.1? 7
5	16.689440	Cisco-L1_aa:63:c2	Intel_46:cf:0d	ARP	192.168.1.1 is at 00:18
6	16.689601	192.168.1.100	203.113.127.199	DNS	Standard query A www.go
7	16.725420	203.113.127.199	192.168.1.100	DNS	Standard query response
8	16.973433	64.233.189.104	192.168.1.100	ICMP	Echo (ping) reply
9	17.921919	64.233.189.104	192.168.1.100	ICMP	Echo (ping) reply
10	18.919756	64.233.189.104	192.168.1.100	ICMP	Echo (ping) reply
11	19.921697	64.233.189.104	192.168.1.100	ICMP	Echo (ping) reply
- Section Packets Header:** Shows the details of the selected packet (Frame 1), including Ethernet II, Internet Protocol, and Internet Group Management Protocol.
- Section Packet Content:** Displays the raw packet data in hexadecimal and ASCII format.

At the bottom, the status bar indicates: Microsoft: <live capture in progress> File: C:/Users/no... | Packets: 11 Displayed: 11 Marked: 0 | Profile: Default

Figure 52 Traffic Monitoring by Wireshark

RESULTS AND DISCUSSION

Results

1. From the field test

In voice transmission, the result of bandwidth monitoring is shown on table 8 (G.711 CODEC), and table 9 (G.729 CODEC). The video transmission monitoring is shown in table 10 (H263 CODEC).

1.1 Voice Transmission using G.711 CODEC

Table 10 Voice Transmission via 2/1 Mbps IPstar Broadband satellite link using G.711 CODEC and SIP Trunk, with VoIP Phone.

Number of Call(s)	CPU Utilize	Bandwidth (Receive/Transmit) KB/s
1	1 %	23.02 / 23.12
2	1 %	34.66 / 46.30
3	1 %	59.45 / 61.64
4	1 %	89.40 / 88.03
5	1 %	107.73 / 105.10
6	1 %	126.62 / 125.42

From the experiment, the voice transmission via IP-Star using G.711 CODEC has bandwidth consumption about 20 kB/s (transmit) per 1 call. From the experiment, when the concurrent calls reaches 10 calls, the bandwidth consumption is about 196.20 kB/s or 1.57 Mb/s. We can determine that the voice transmission via 2/1 Mbps (2 Mbps download/1 Mbps upload) IP-Star broadband using G.711 CODEC can support up to 6 calls for total transmission (incoming and outgoing).

Notice: without the GUI Graphic Interface, the CPU utilize is very little.

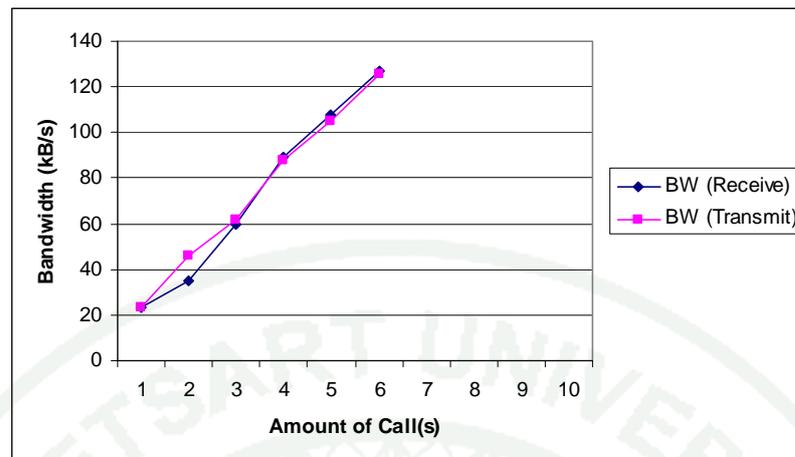


Figure 53 The statistics graph from the voice transmission via 2/1 Mbps IPstar Broadband satellite link using G.711 CODEC and SIP Trunk

1.2 Voice Transmission using G.729 CODEC

Table 11 Voice Transmission via 2/1 Mbps IPstar Broadband satellite link using G.729 CODEC and SIP Trunk, with VoIP Phone.

Number of Call(s)	CPU Utilize	Bandwidth (Receive/Transmit) KB/s
1	1 %	14.63 / 14.38
2	1 %	29.05 / 28.99
3	1 %	40.76 / 43.33
4	1 %	47.01 / 51.28
5	1 %	62.37 /66.27
6	1 %	73.43 /78.12
7	2 %	88.43 /89.26
8	2 %	97.34 /98.25
9	2 %	109.03 /114.32
10	3 %	122.43 / 125.27

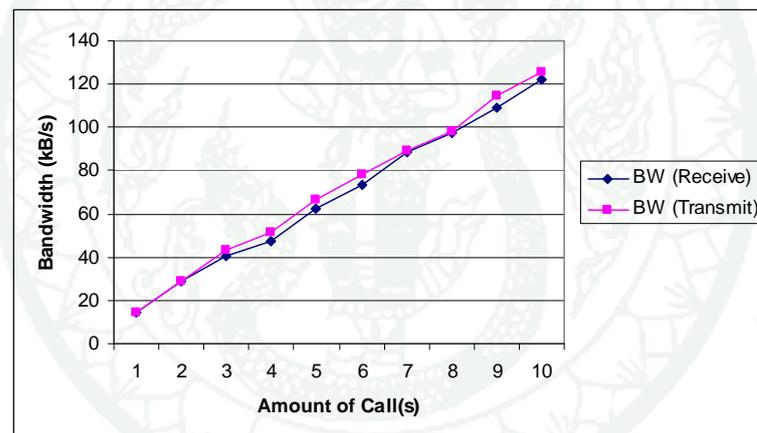


Figure 54 The statistics graph from the voice transmission via 2/1 Mbps IPstar Broadband satellite link using G.729 CODEC and SIP Trunk

From the experiment, the voice transmission via IP-Star using G.729 CODEC has bandwidth consumption about 15 kB/s (transmit) per 1 call. From the experiment, when the concurrent calls reaches 10 calls, the bandwidth consumption is about 125.27 kB/s or 1.0 Mb/s. We can determine that the voice transmission via 2/1 Mbps IP-Star

broadband using G.729 CODEC can support up to 10 calls for total transmission (incoming and outgoing). The bandwidth consumption of 10 calls is about 1002 kb/s or 1.0 Mbps.

Notice: without the GUI Graphic Interface, the CPU utilize is very little.

1.3 Video Transmission using H.263 codec

Table 12 Video Transmission via 2/1 Mbps IPstar Broadband satellite link using H.263 CODEC and SIP Trunk (software: eyebeam 1.5)

Number of Call(s)	CPUs	Bandwidth (Receive/Transmit) KB/s
1	1 %	113.97 / 134.83

From the experiment, the video transmission via IP-Star using H.264 CODEC a bandwidth consumption about 135 kB/s(Transmit) or 1080 kbps (or 1 Mbps) per 1 call. The network bandwidth consumption per 1 call is about 0.5 Mbps. However, in video transmission, in the same way of voice transmission, the CODEC selection is the main factor in call management. The software for video call is also important issue, some version gives a high resolution but also high bandwidth consumption.

It is clear that CODECs with lower bit rates shall result in more number of simultaneous video calls.

In case of jitterbuffer setting, the end-to-end measurement delay would be increased in video transmission.

1.4 Voice and video Transmission using G.711, G.729 and H.263 (video)codec

We first test the G.711 , G.729 and H.263 CODEC on voice and video transmission using both SoftPhone and VoIP Phone, and then test for two CODEC

coordination (G.711 and G.729, Transcoder) where one side use G.711, other side use G.729, with no silent suppression (Voice and video Transmission via 2/1 Mbps IPStar Broadband satellite link and IAX2 Trunk)



Figure 55 The voice transmission by VoIP Phone using G.711

Transmission rate = 9.8 kB/s
Received rate = 9.0 kB/s

Src IP addr	Src port	Dest IP addr	Dest port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)	Pb?
192.168.66.201	48106	192.168.66.100	7140	0x69A52853	ITU-T G.711 PCMA	3872	-1936 (-100.0%)	34.29	2.59	1.09	X
192.168.66.201	4392	192.168.66.100	7142	0xA0C5F0D0	Unknown(124)	12	-8 (-200.0%)	10179.96	1686.87	2544.90	X
192.168.66.100	7140	192.168.66.201	48106	0x562D5581	ITU-T G.711 PCMA	1632	8 (0.5%)	231.99	31.87	16.93	X

Figure 56 The voice transmission by VoIP Phone using G.711 CODEC

Average Fwd. Jitter = 1.09 ms
Average Rev. Jitter = 16.93 ms
Average Fwd. Packet Loss = 0.5 %
Average Rev. Packet Loss = 0.5 %



Figure 57 The video transmission by SoftPhone using G.711 and H.263

Transmission rate = 49.9 kB/s
Received rate = 23.0 kB/s

Src IP addr	Src port	Dest IP addr	Dest port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)	Pb?
192.168.66.201	37886	192.168.66.100	6990	0xD5FC8BB5	ITU-T G.711 PCMA	4227	-2113 (-100.0%)	33.71	2.36	1.19	X
192.168.66.201	12146	192.168.66.100	6992	0x3566986D	Unknown(126)	1982	-993 (-100.4%)	6461.42	396.56	343.76	X
192.168.66.100	6990	192.168.66.201	37886	0xC9003719	ITU-T G.711 PCMA	1965	2 (0.1%)	164.28	27.09	15.33	X
192.168.66.100	6992	192.168.66.201	12146	0xB5F8BCF	H263-1998	513	4 (0.8%)	406.08	56.46	4.34	X

Figure 58 The video transmission by SoftPhone using G.711 and H.263 codec

Average Fwd. Jitter = 1.19 ms
 Average Rev. Jitter = 19.67 ms
 Average Fwd. Packet Loss = 0.9 %
 Average Rev. Packet Loss = 0.9 %

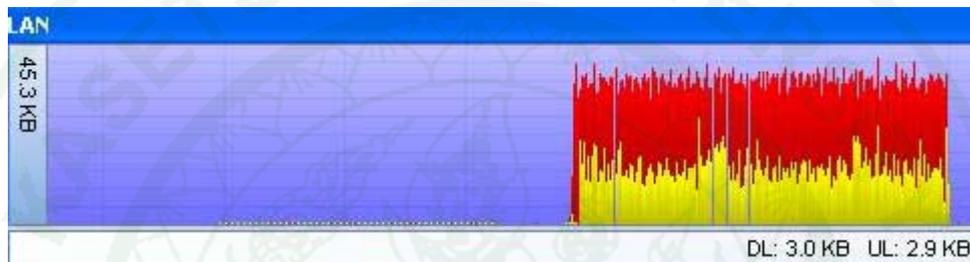


Figure 59 The voice transmission by VoIP Phone using G.729 CODEC

Transmission rate = 2.9 kB/s
 Received rate = 3.0 kB/s

Src IP addr	Src port	Dest IP addr	Dest port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)	Pb?
192.168.66.201	43782	192.168.66.100	7660	0xE6869319	ITU-T G.729	6037	-3018 (-100.0%)	31.74	2.13	0.97	X
192.168.66.201	37438	192.168.66.100	7662	0x8DD0C2E7	Unknown(124)	16	-10 (-166.7%)	10461.64	2566.90	3268.21	X
192.168.66.100	7660	192.168.66.201	43782	0x1FAF2927	ITU-T G.729	142	0 (0.0%)	80.59	13.34	10.72	X
192.168.66.100	7660	192.168.66.201	43782	0x613B2C29	ITU-T G.729	2591	3 (0.1%)	197.96	31.44	12.45	X

Figure 60 The voice transmission by VoIP Phone using G.729 CODEC

Average Fwd. Jitter = 0.97 ms
 Average Rev. Jitter = 12.45 ms
 Average Fwd. Packet Loss = 0.1 %
 Average Rev. Packet Loss = 0.1 %



Figure 61 The video transmission by SoftPhone using G.729 and H.263

Transmission rate = 41.8 kB/s
 Received rate = 13.1 kB/s

Src IP addr	Src port	Dest IP addr	Dest port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)	Pb? -
192.168.66.201	51060	192.168.66.100	7610	0x588C82C4	ITU-T G.729	6050	-3025 (-100.0%)	35.73	4.57	1.30	X
192.168.66.201	4556	192.168.66.100	7612	0xD0D5F9BA	Unknown(126)	3308	-1656 (-100.2%)	1170.50	60.47	57.49	X
192.168.66.100	7610	192.168.66.201	51060	0xD23CBAE3	ITU-T G.729	2886	3 (0.1%)	201.69	33.22	14.94	X
192.168.66.100	7612	192.168.66.201	4556	0x41E3E382	H263-1998	522	3 (0.6%)	445.39	103.80	18.58	X

Figure 62 The video transmission by SoftPhone using G.729 and H.263 codec

Average Fwd. Jitter = 1.30 ms
 Average Rev. Jitter = 33.52 ms
 Average Fwd. Packet Loss = 0.7 %
 Average Rev. Packet Loss = 0.7 %

Table 13 Characteristics of voice transmission based on terminal types and CODECs

Codec	Terminal Type	Bandwidth (voice only)		Jitter		Packet Loss	
		DL (KB/s)	UL (KB/s)	Fwd. (ms)	Rev. (ms)	Fwd. (%)	Rev. (%)
G.711	VoIP Phone	10.2	10.1	1.09	16.93	0.5	0.5
G.711	SoftPhone	9.0	9.8	0.98	12.94	16.3	16.3
G.711	SoftPhone (voice)			1.19	19.67	0.9	0.9
G.711-G.729*	VoIP Phone	3.5	3.3	0.8	11.96	7.9	7.9
G.711-G.729	SoftPhone	3.0	2.9	0.63	5.70	0	0
G.711-G.729*	SoftPhone (voice)			0.69	20.49	0.7	0.7
G.711-G.729*	SoftPhone (video)						
G.729	VoIP Phone	3.5	3.3	0.97	12.45	0.1	0.1
G.729	SoftPhone	3.0	2.9	1.16	12.22	0.1	0.1
G.729	SoftPhone (voice)			1.30	33.52	0.7	0.7
G.729	SoftPhone (video)						

* one side use G.711, other side use G.729

From this experiment, we learned that:

VoIP Phone (hardware terminal) requires a little bit more bandwidth because CODEC is implemented as embedded system which is less efficient compared to software-based VoIP phone. The latter runs on a more powerful computer platform. Lower bandwidth requirement generally results in lower jitter and packet loss. Exception is when making a video call which video packets exacerbate this result (The bit rate supported for H.263 CODEC is a variable bit rate (VBR), Each manufacturer customizes its H.263 profile to suit its needs).

G.729 yields lower voice bandwidth. However, our experiments is based on only one call therefore both CODEC did not saturate the link and, therefore, the jitter and packet loss were not much different.

For different CODECs at each end (G.711-G.729), G.729 overrules.

In video transmission, there were a number of jitter, and it effected to a quality of voice and a quality of the connection.

A typical jitter buffer configuration is 30mS to 50mS in size. In the case of video transmission jitter buffer then the maximum size may be set to 100-500mS. To resolved this problem, we went to the jitterbuffer configuration, by approximately value from 500 ms and varied (by approximation) based on the packet loss and delay time monitored from Netstat screen.

Description:

- RTT: round-trip time ; the time required to send a signal in both directions (milliseconds)
- Jit: The jitter we have measured (milliseconds)
- Del: The maximum delay imposed by the jitterbuffer (milliseconds)
- Lost: The number of packets we've detected as lost.
- %: The percentage of packets we've detected as lost recently.
- Drop: The number of packets we've purposely dropped (to lower latency).
- OOO: The number of packets we've received out-of-order
- Kpkts: The number of packets we've received / 1000.

```

----- LOCAL -----
Channel      RTT  Jit  Del  Lost  %  Drop  000  Kpkts  Jit  Del  Lost  %  Drop  000  Kpkts  FirstMsg  LastMsg
IAX2/MainCenter-176  703 2178 2000 1048 57 43012 15395 87 131 182 1089 3 90 26017 59 Tx:NEW Tx:LAGRQ
IAX2/field_1-user-4328 680 659 717 66 1 2611 24650 148 135 187 3907 4 445 644 141 Rx:NEW Tx:LAGRP
2 active IAX channels

```

Figure 63 The video transmission by SoftPhone using G.711 and H.263 CODEC (no jitter buffer)

```

----- LOCAL -----
Channel      RTT  Jit Del  Lost  % Drop  000  Kpkts  Jit Del  Lost  % Drop  000  Kpkts  FirstMsg  LastMsg
IAX2/field_1-user-4083 642  97  571    2    0    30 1933    5 412 488  77    6    5 288    3 Rx:NEW  Tx:ACK
1 active IAX channel

```

Figure 64 The video transmission by SoftPhone using G.711 and H.263 CODEC with Jitterbuffer configuration

After the jitterbuffer size was increased, number of packet loss and drop packet was reduced. But a number of delay still occurred.

```

-- G.729 PLC
localhost*CLI> iax2 show netstats
----- LOCAL -----
Channel      RTT  Jit Del  Lost  % Drop  000  Kpkts  Jit Del  Lost  % Drop  000  Kpkts  FirstMsg  LastMsg
IAX2/MainCenter-176 1871 124 672    7    1   261 3065    7  84 153  94    2    8 729    5 Tx:NEW  Rx:ACK
IAX2/field_1-user-4328 660  100 701   41    1   173 23355   120  79 145 2905  2   350 522   115 Rx:NEW  Tx:ACK
2 active IAX channels

```

Figure 65 The video transmission by SoftPhone using G.729 and H.263 CODEC (no jitter buffer)

```

----- LOCAL -----
Channel      RTT  Jit Del  Lost  % Drop  000  Kpkts  Jit Del  Lost  % Drop  000  Kpkts  FirstMsg  LastMsg
IAX2/field_1-user-299 609  430 722    2    2    64 3284    9  95 175 121    4   16 351    5 Rx:NEW  Tx:ACK
1 active IAX channel

```

Figure 66 The video transmission by SoftPhone using G.729 and H.263 CODEC with Jitterbuffer configuration

After the jitterbuffer size was increased, number of packet loss and drop packet was reduced. But a number of delay still occurred.

By jitterbuffer configuration, we can reduce a packet loss and get a better quality in transmission in both voice and video transmission. The notification is, jitterbuffer will increase the delay of end-to-end transmission. Therefore, with clearer voice, the delay time will be expanded.

From the experiment, jitter buffer is recommended for video transmission where we can less aware of frame delay or more acceptable when compare with the percent of video traffic dropped. On the other hand, for emergency situation, the video

transmission may not be constrained to meet a hard real-time transmission. The actual need is to monitor the current environment.

From the simulation results, according to system's response, we can determine that the system can support the simultaneous calls at the maximum rate of 160 call/s for voice traffic, and 150 call/s for video traffic. At this condition, the system can work in function properly, therefore, the percent of call fail is very little and the ability in traffic management such as the call setup and call clear time are processed within acceptable value.

The maximum rate call is the useful information in providing the internal call management or internal devices registration (Internal call means: the communication in one network base station without the satellite link or the link with other networks). And also be used in considering for the capability of the future communication, e.g. 75 Mbps WiMAX Technology, instead of 2/1 Mbps IPStar Broadband, for the next generation of broadband communication in Thailand.

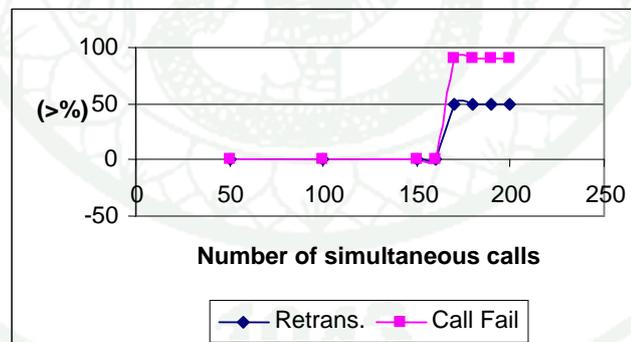


Figure 67 Call Retransmissions and Call Fail from voice traffic simulations

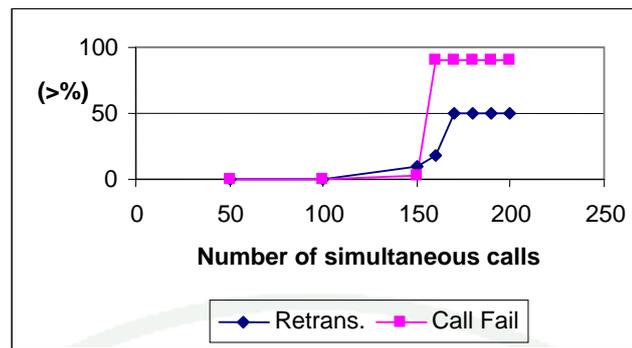
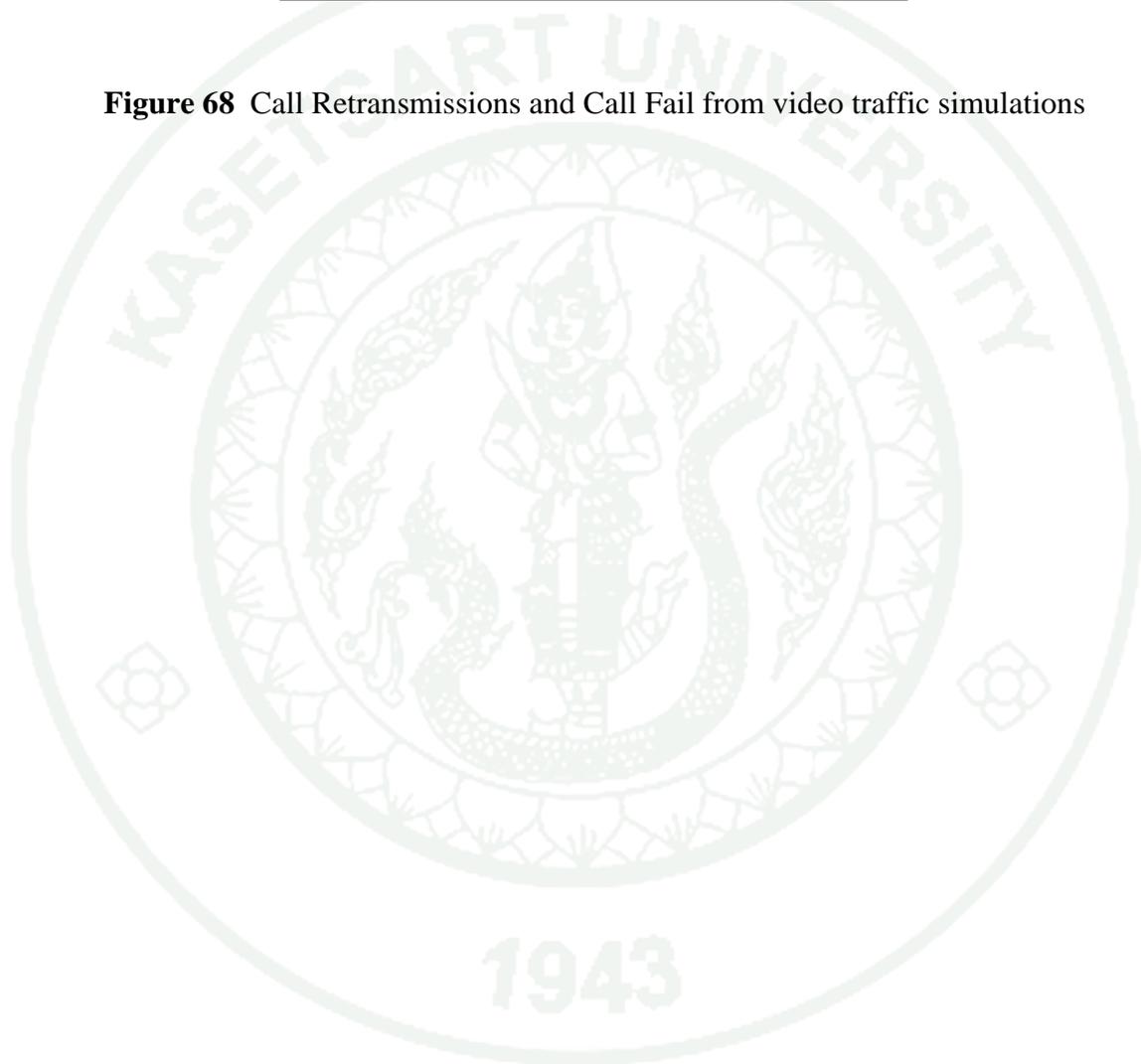


Figure 68 Call Retransmissions and Call Fail from video traffic simulations



Discussions

1. The selection of VoIP software solution may affect system performance in the following ways:

a. The number of simultaneous call, this can be changed by tweaking some parameters in the software.

b. The supported CODECs, this can be selectively and temporarily disabled to maintain QoS of the system.

c. The software itself may be customized to match the available computing resources to maintain system performance.

2. In order to effectively employ RoIP on top of VoIP system, some hardware development may be required. For example, the developed VOX enhances and Telephone Hybrid unit.

3. Good system configuration for emergency communication should give priority to voice quality¹, video quality, and number of simultaneous calls, respectively. Our experiments show that G.711 voice CODEC is more preferred than G.729. By the way,

¹ Voice quality is measured by MOS (Mean Opinion Score),
http://en.wikipedia.org/wiki/Mean_opinion_score

CONCLUSION AND RECOMMENDATION

Conclusion

1. To support the good traffic management for emergency link via IPStar, the codec selection is the most important to be considered. We should firstly consider the supported codec for the system, and then consider the supported communication devices.

2. In software simulation, the dial plan (call scenarios) is one important issue for the best performance. The good dial plan configuration that uses the shortest call set up time can reduce the number of call retransmission and call failed. To find a good dial plan, we should consider the supported codec, the priority extension, time used, and call loop.

3. The limitation of the system is depended on three factors, the computing resources (CPUs utilized and RAM), IP-PBX Software, and System Configuration (Dial Plan, Jitter Buffer and Delay).

4. In Jitter Buffer Configuration, the approximate value to be used in configure is considered from the real environment test. The suitable value measured from the call in both voice and video transmission for each codec can be used as an initial value for the next configuration.

Recommendation

1. In CODEC selection, we should consider from bandwidth capacity and the priority of communication. From our experiment, in case the need of concurrent calls is the first priority, the suitable CODEC is G.729 (support up to 10 simultaneous calls at 2/1 Mbps). But in case we need the voice quality, G.711 CODEC is preferred (but it support up to 6 simultaneous calls at 2/1 Mbps).

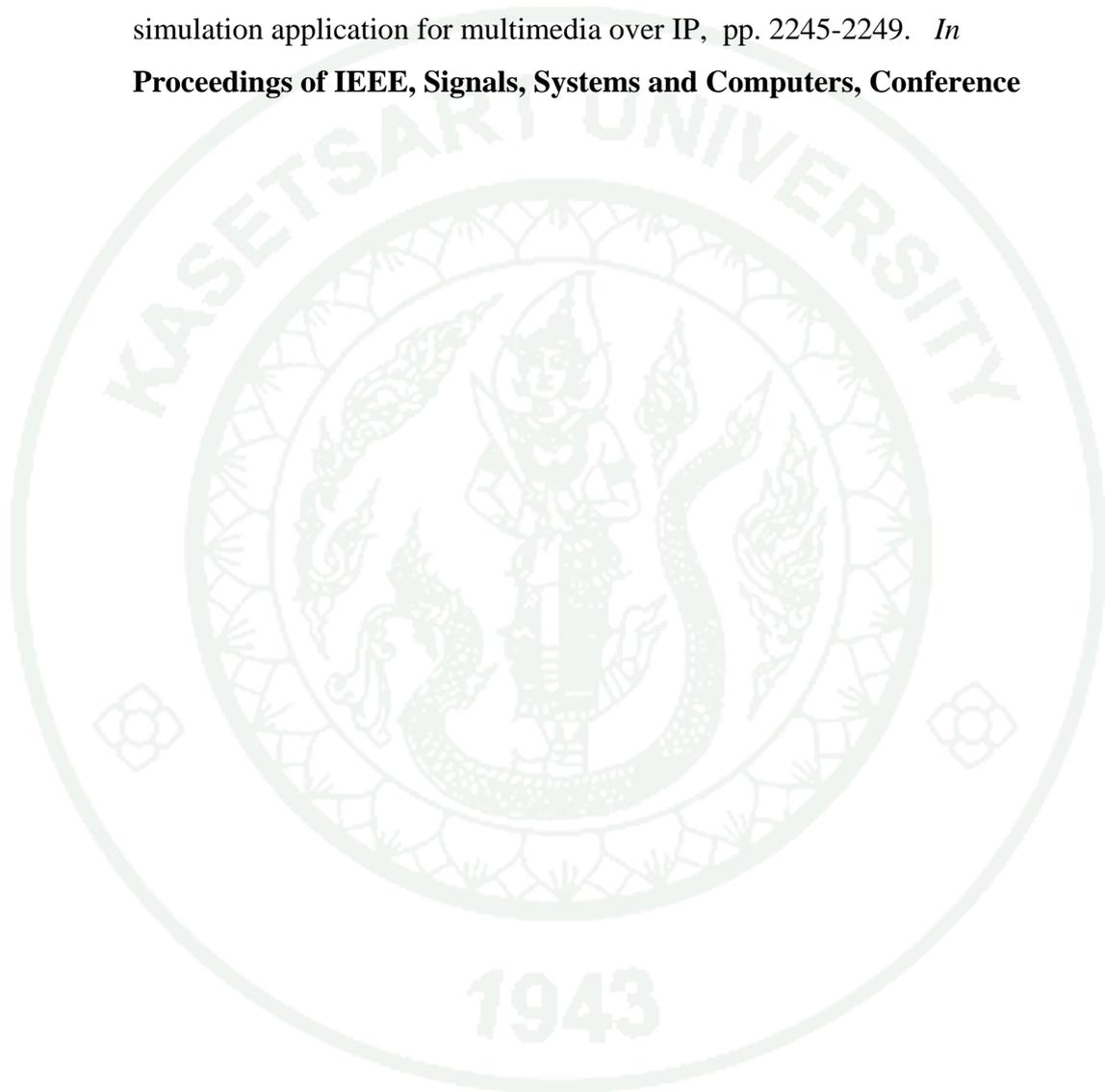
2. In jitter buffer configuration, it certainly concern with the end to end delay. The more jitter buffer size, the more delay get from the system. The suitable values should be observed from the experiments. We may consider from the priority of use. In case we need the quality of voice (the end to end delay of voice transmission is not exceed 1 s) the size of jitter buffer should be approximately 30-50 ms. If we need the quality of video transmission, the size of jitter buffer should be bigger (100-500 ms).

LITERATURE CITED

- Castro Marcel C., Peter Dely, Jonas Karlsson and Andreas Kessler, "Capacity Increase for Voice over IP Traffic through Packet Aggregation in Wireless Multihop Mesh Networks", **IEEE Future Generation Communication and Networking (FGCN 2007)**, Volume 2, issue 6-8. 350 – 355, Dec. 2007
- Crisis Communication Unit, Health Canada. 2003. **Crisis/Emergency Communications Guidelines**. 25.
- ITU-T. 2004. **G.711 Pulse code modulation (PCM) of voice frequencies Recommendation**.
- ITU-T. 2005. **H.264 Advanced video coding for generic audiovisual services**.
- ITU-T. 2005. **H.264 SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services**
- Lin, J.-R., Pang, A.-C., and Wang, Y.-C., "iPTT: Peer-to-Peer Push-to-Talk for VoIP", **Journal of Wireless Communications and Mobile Computing (WCMC)**, **8(10)**.1331-1343, 2008.
- Raytheon Company White Paper: WP-3004-2. **RoIP Versus VoIP Solutions**.
http://www.raytheon.com/capabilities/rtnwcm/groups/ncs/documents/content/rtn_ncs_products_nxu2a9_pdf.pdf
- RFC 3261 2002. **Standards Track SIP: Session Initiation Protocol**
- VoIP-Info.org. **Asterisk ZAP channels**. <http://www.voip-info.org/wiki/index.php?page=Asterisk+Zap+channels>.

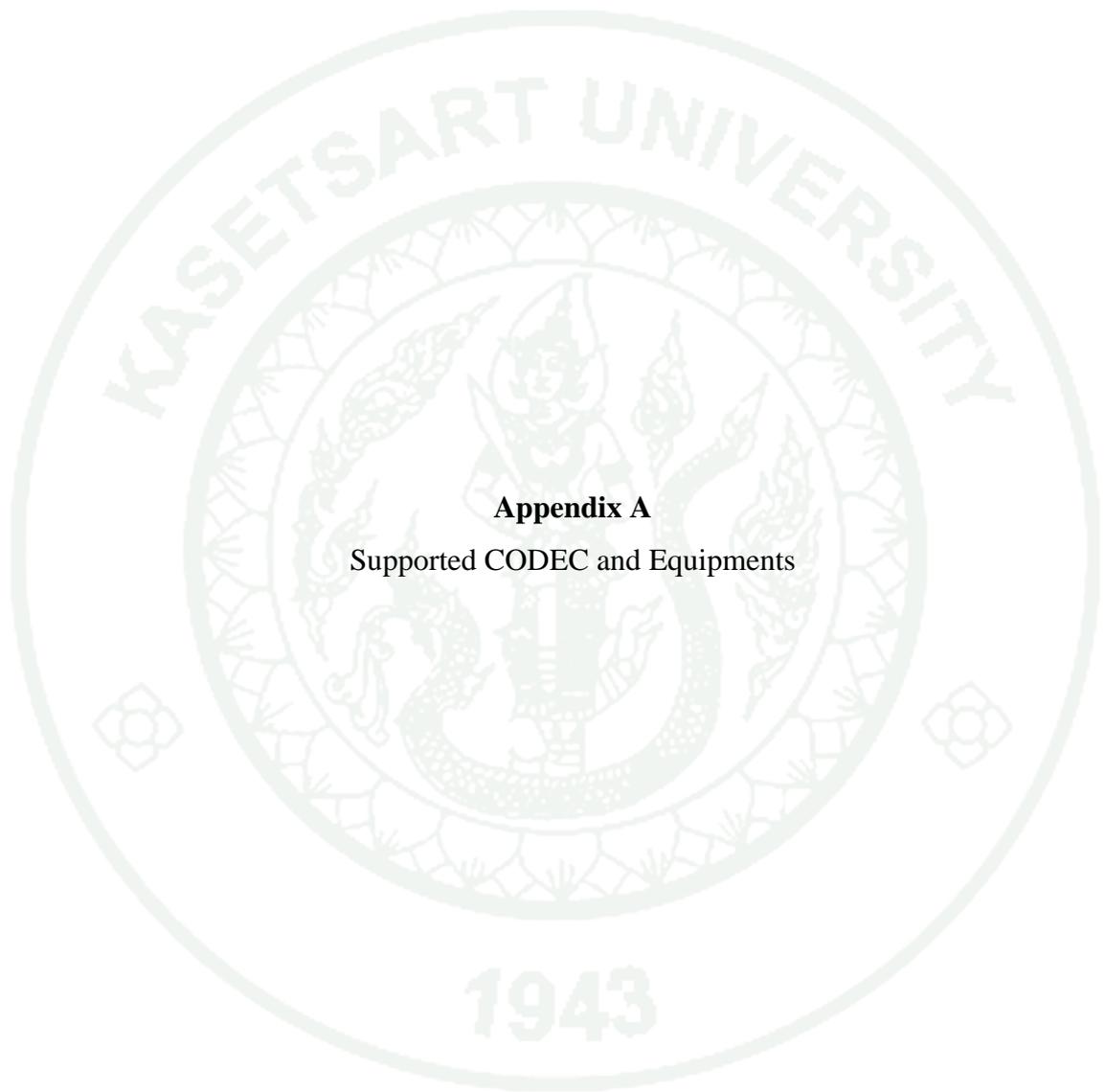
Walker John Q., NetIQ Corporation White Paper. 2005. **Assessing VoIP Call Quality Using the E-model**. http://www.ixiacom.com/pdfs/library/white_papers/voip_quality.pdf

White, Chirstopher M., Josh Raymond and Keith A. 2004. A real timenetwork simulation application for multimedia over IP, pp. 2245-2249. *In Proceedings of IEEE, Signals, Systems and Computers, Conference*





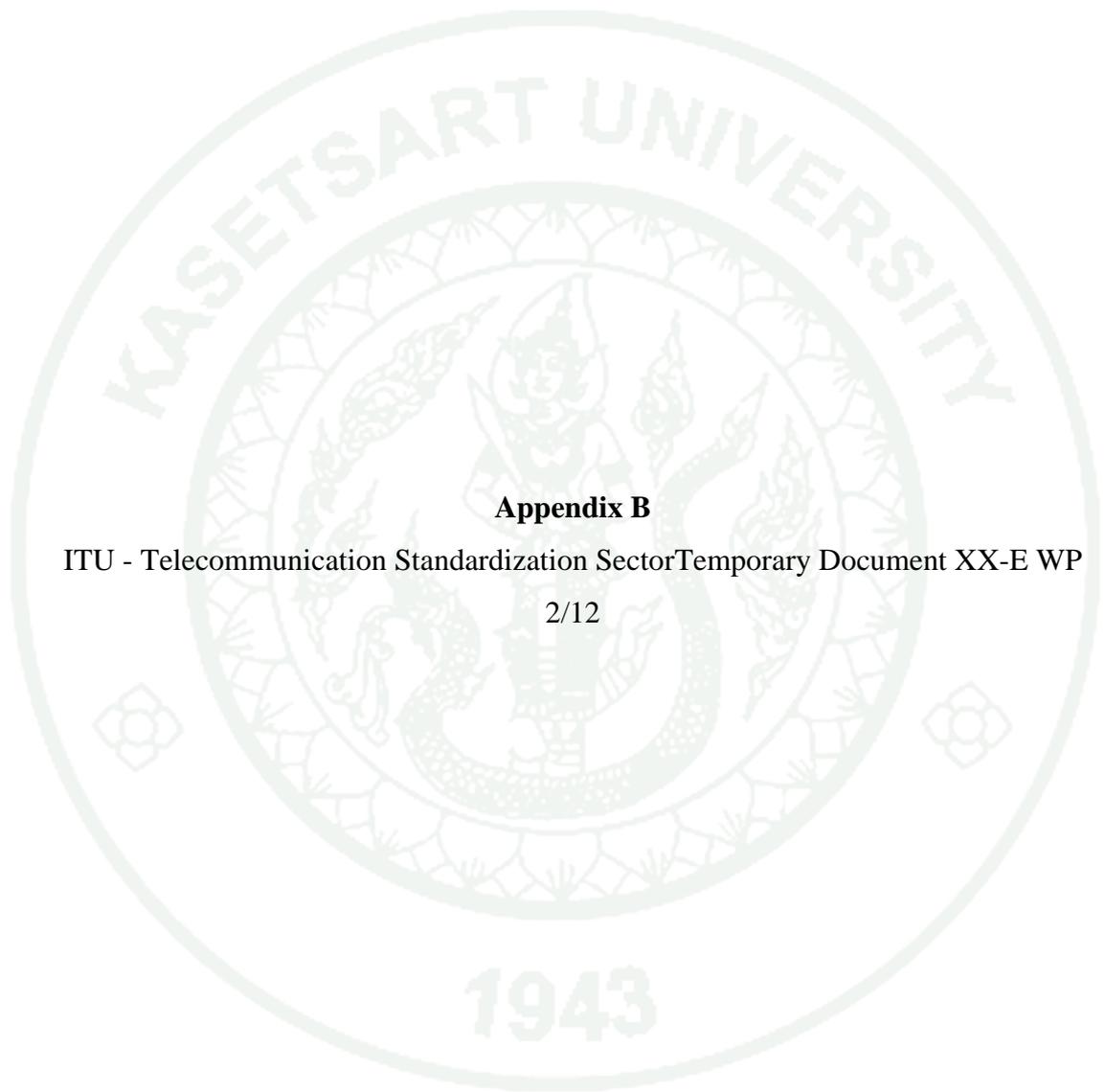
APPENDICES



Appendix A

Supported CODEC and Equipments

Client	Supported CODEC
Billion BIPAC 7402VL	G.711, G.729
Billion BIPAC-7100SV	G.711, G.729
Billion BIPAC7402VGP	G.711, G.729
Cisco 7960	G.711, G.729
Draytek Vigor 2100V(G)	G.711, G.723.1, G.726, G.729
Draytek Vigor 2500V	G.711, G.729
Draytek Vigor 2600V(G)	G.711, G.723.1, G.726, G.729
Draytek Vigor 2900V(G)	G.711, G.723.1, G.726, G.729
eyeBeam	GSM, iLBC, G.711, G.722, G.723.1, G.729, Speex
Grandstream BudgeTone 101	iLBC, G.711, G.723.1, G.726, G.728, G.729
Grandstream BudgeTone 102	iLBC, G.711, G.723.1, G.726, G.728, G.729
Grandstream GXP2000	GSM, G.711, G.722, G.723.1, G.726, G.728, G.729
Grandstream HandyTone 286	iLBC, G.711, G.723.1, G.726, G.728, G.729
Grandstream Handytone 486	iLBC, G.711, G.723.1, G.726, G.728, G.729
Leadtek 8051	G.711, G.723.1, G.726, G.729
Linksys PAP2	G.711, G.723.1, G.726, G.729
Linksys RT31P2	G.711, G.723.1, G.726, G.729
Linksys WRT54GP2	G.711, G.729
MS Office Communicator	GSM, G.711, G.722, G.723.1, DVI4, Siren
Octtel SPxxxx Series Gateways	G.711, G.729
Polycom SoundPoint IP300	G.711, G.729
Polycom SoundPoint IP500	G.711, G.729
Polycom SoundPoint IP600	G.711, G.729
Siemens optiPoint 400 Family	G.723.1
Siemens optiPoint 410 Family	G.711, G.723.1
Siemens optiPoint 420 Family	G.711, G.722, G.723.1, G.729
Sipura SPA-2000	G.711, G.723.1, G.726, G.729
Sipura SPA-2100	G.711, G.723.1, G.726, G.729
Sipura SPA-3000	G.711, G.723.1, G.726, G.729
Sipura SPA-841	G.711, G.729
sipXphone	G.711
SJPhone (free version)	GSM, iLBC, G.711
Snom 190	GSM, G.711, G.722, G.723.1, G.726, G.729
Snom 320	GSM, G.711, G.722, G.723.1, G.726, G.729
Snom 360	GSM, G.711, G.722, G.723.1, G.726, G.729
SwissVoice IP 10S	iLBC, G.729
Uniden UIP-200	G.711, G.729
Windows Messenger	GSM, G.711, G.722, G.723.1, DVI4, Siren
X-lite	GSM, iLBC, G.711, Speex
X-Pro	GSM, iLBC, G.711, G.729, Speex
Zyxel Prestige 2000W	G.711, G.729
Zyxel Prestige 2002	G.711, G.729
Zyxel Prestige 2602HW(-L)	G.711, G.729



Appendix B

ITU - Telecommunication Standardization Sector Temporary Document XX-E WP

2/12

ITU - Telecommunication Standardization Sector Temporary Document XX-E WP 2/12

STUDY GROUP 12

Geneva, 27-31 May 2002

Question(s): 8/12

SOURCE*: RAPPORTEUR FOR QUESTION 8/12

TITLE: INFORMATION ABOUT A NEW METHOD FOR DERIVING THE
TRANSMISSION RATING FACTOR R FROM MOS IN CLOSED FORM

This document, for information, presents a new method for deriving the transmission rating factor R from MOS-values in closed form using a slightly modified version of the transformation formula to be found in ITU-T Rec. G.107 (2000). The formula was found and brought to our attention by Mr. Christian Hoene from the Telecommunication Networks Group at the Technical University Berlin, Germany. It is addressed to Question 8 in order to decide whether it will be useful to include such a formula in future versions of G.107, e.g. as an Appendix which might be useful for applying Rec. P.833 and draft new Rec. P.DIEM.

Introduction

The E-Model (ITU-T Rec. G.107) is the recommended computational model to be used for transmission-planning. The E-Model predicts the impact of typical impairments on speech quality. The primary output is the "Rating Factor" R . It ranges from 0 (worst) to 100 (best). The Mean Opinion Score (MOS) can be obtained from the R Factor with a *converting formula*. MOS has a scale from one (bad) to five (excellent) and is used for subjective and objective evaluations of perceived speech quality.

In this report we describe an unexpected behavior of the converting formula given in G.107 for low R values. As a consequence, the formula cannot be inverted to calculate the R factor from a given MOS value. Thus, we redefine the converting formulae and present its inverted form, which allows R factors to be calculated from MOS values.

* Contact:	Sebastian MOELLER (Deutsche Telekom)	Tel:	+49 234 32 23979
	Institute of Communication Acoustics	Fax:	+49 234 32 14165
	Ruhr-University Bochum	E-mail:	moeller@ika.ruhr-uni-bochum.de

Attention: This is not a publication made available to the public, but an **internal ITU-T Document** intended only for use by the Member States of the ITU, by ITU-T Sector Members and Associates, and their respective staff and collaborators in their ITU related work. It shall not be made available to, and used by, any other persons or entities without the prior written consent of the ITU-T.

ITU-T\COM-T\COM12\TD\WP2\009E

ITU - Telecommunication Standardization Sector Temporary Document XX-E WP 2/12

STUDY GROUP 12

Geneva, 27-31 May 2002

Question(s): 8/12

SOURCE*: RAPPORTEUR FOR QUESTION 8/12

TITLE: INFORMATION ABOUT A NEW METHOD FOR DERIVING THE
TRANSMISSION RATING FACTOR R FROM MOS IN CLOSED FORM

This document, for information, presents a new method for deriving the transmission rating factor R from MOS-values in closed form using a slightly modified version of the transformation formula to be found in ITU-T Rec. G.107 (2000). The formula was found and brought to our attention by Mr. Christian Hoene from the Telecommunication Networks Group at the Technical University Berlin, Germany. It is addressed to Question 8 in order to decide whether it will be useful to include such a formula in future versions of G.107, e.g. as an Appendix which might be useful for applying Rec. P.833 and draft new Rec. P.DIEM.

Introduction

The E-Model (ITU-T Rec. G.107) is the recommended computational model to be used for transmission-planning. The E-Model predicts the impact of typical impairments on speech quality. The primary output is the "Rating Factor" R . It ranges from 0 (worst) to 100 (best). The Mean Option Score (MOS) can be obtained from the R Factor with a *converting formula*. MOS has a scale from one (bad) to five (excellent) and is used for subjective and objective evaluations of perceived speech quality.

In this report we describe an unexpected behavior of the converting formula given in G.107 for low R values. As a consequence, the formula cannot be inverted to calculate the R factor from a given MOS value. Thus, we redefine the converting formulae and present its inverted form, which allows R factors to be calculated from MOS values.

* Contact:	Sebastian MOELLER (Deutsche Telekom)	Tel:	+49 234 32 23979
	Institute of Communication Acoustics	Fax:	+49 234 32 14165
	Ruhr-University Bochum	E-mail:	moeller@ika.ruhr-uni-bochum.de

Attention: This is not a publication made available to the public, but an **internal ITU-T Document** intended only for use by the Member States of the ITU, by ITU-T Sector Members and Associates, and their respective staff and collaborators in their ITU related work. It shall not be made available to, and used by, any other persons or entities without the prior written consent of the ITU-T.

ITU-T\COM-T\COM12\TD\WP2\009E

- 2 -

New Conversion Formula

The Mean Option Score (MOS) can be obtained from the R-Factor as described in ITU-T Rec. G.107 (2000) (Equation B-4).

$$MOS = \begin{cases} R \leq 0: & 1 \\ 0 \leq R \leq 100: & 1 - \frac{7}{1000}R + \frac{7}{6250}R^2 - \frac{7}{1000000}R^3 \\ R \leq 100: & 4.5 \end{cases} \quad (1)$$

This formula calculates for each R -value between 0 and $80 - 30\sqrt{6} \approx 6,5$ an MOS value below 1 (see

Figure 1). Thus, the formulae cannot be inverted (as stated also in ITU-T Rec. P.833, 2001) to calculate - even within the given range of R and MOS - one R factor from a given MOS value.

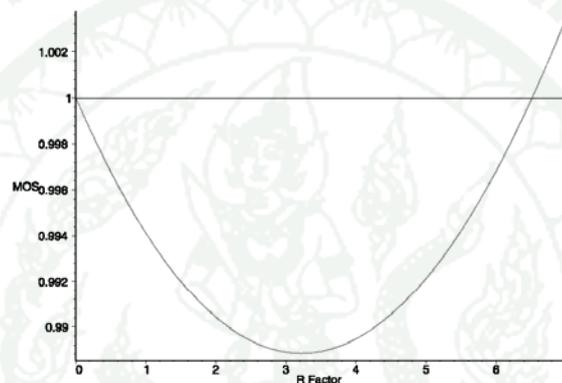


Figure 1: MOS as function of rating factor R

Therefore, we redefine the converting formula to:

$$MOS = \begin{cases} R \leq 6.5: & 1 \\ 6.5 \leq R \leq 100: & 1 - \frac{7}{1000}R + \frac{7}{6250}R^2 - \frac{7}{1000000}R^3 \\ R \leq 100: & 4.5 \end{cases} \quad (2)$$

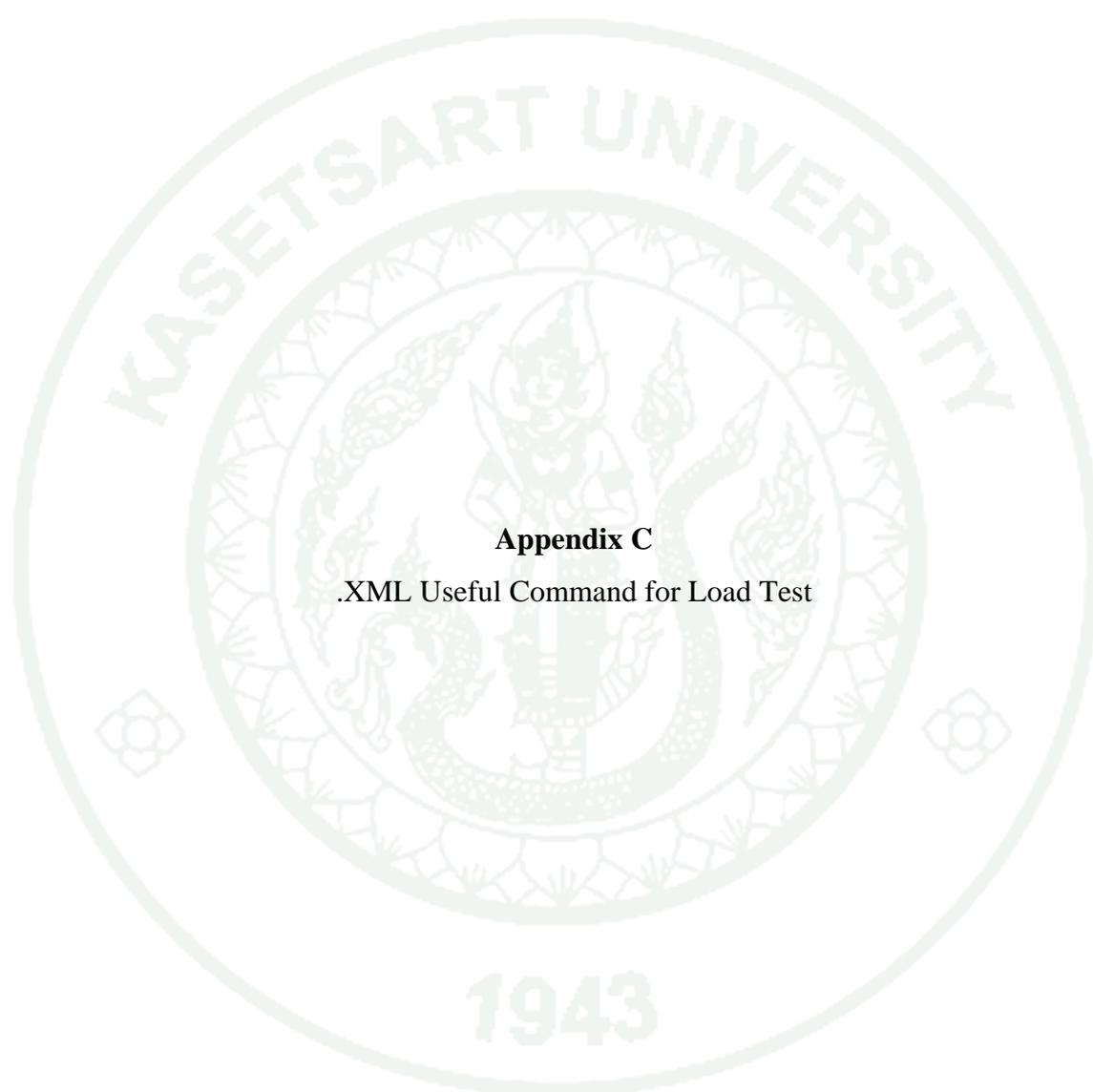
This equation can be inverted with the Candono's Formula:

$$R = \frac{160}{3} - \frac{5}{21} \left((1+i\sqrt{3})\sqrt[3]{p+iq} + (1+i\sqrt{3})\sqrt[3]{p-iq} \right) \quad (3)$$

with the MOS range between 1 and 4.5 and

$$\begin{aligned} q &= 1470\sqrt{-903522 + 1113960 \text{ MOS} - 202500 \text{ MOS}^2} \\ p &= 1819468 - 661500 \text{ MOS} \\ p, q &> 0 \end{aligned} \quad (4)$$

The equation requires computations with complex numbers. However, the result R is a real value in any case. Thus, we simplify the term to:



Appendix C
.XML Useful Command for Load Test

.XML Useful Command for Load Test

- *-sf*, loads an alternate scenario file
- *-inf*, loads an external CSV file to inject values from
- *-r*, sets the call rate (calls per second)
- *-rp*, specify the rate period (ms) for the call rate (above)
 - Example: *-r 15 -rp 1000*, 15 calls every second
- *-l*, maximum number of simultaneous calls
- *-m*, stop and exit when set number of calls are processed
- *-i*, sets the local IP address
- *-p*, sets the local port number
- *-auth_uri*, forces the domain value of the URI for authentication (for example, 401)
- *-trace_screen*, logs the final screens to file (<scenario>_<pid>_screen.log)
- *-trace_err*, logs unexpected messages to file (<scenario>_<pid>_errors.log)
- *-trace_stat*, logs statistics to file (<scenario>_<pid>.csv)
- *-fd*, logs statistics at set interval (seconds)
- *-nd*, disable SIPp's default behaviors
- Finally, we end all SIPp commands with the remote host (and port), for example the P-CSCF

Creating Scenarios

- Useful elements:
 - <send>
 - <recv>
 - <pause>, speech time
 - <nop>, no operation
 - <label>

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">

<scenario name="UAC Basic Registration">
  <label id="0" />
  <send retrans="500" start_rtd="true">
    <![CDATA[
      REGISTER sip:[field1] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
      From: [field0] <sip:[field0]@[field1]>;tag=[call_number]
      To: [field0] <sip:[field0]@[field1]>
      Call-ID: [call_id]
      CSeq: [cseq] REGISTER
      Contact: [field0] <sip:[field0]@[local_ip]:[local_port]>;expires=691200
```

```

Max-Forwards: 70
Content-Length: 0
]]>
</send>

<recv response="200" rtd="true" next="3" optional="true" />
<recv response="403" rtd="true" next="2" optional="true" />
<recv response="404" rtd="true" next="2" optional="true" />
<recv response="401" rtd="true" next="1" auth="true" crlf="true" />

<label id="1" />
<send retrans="500" start_rtd="true">
  <![CDATA[
    REGISTER sip:[field1] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: [field0] <sip:[field0]@[field1]>;tag=[call_number]
    To: [field0] <sip:[field0]@[field1]>
    Call-ID: [call_id]
    CSeq: [cseq] REGISTER
    Contact: [field0] <sip:[field0]@[local_ip]:[local_port]>;expires=691200
    [field2]
    Max-Forwards: 70
    Content-Length: 0
  ]]>
</send>

<recv response="400" rtd="true" next="2" optional="true" />
<recv response="403" rtd="true" next="2" optional="true" />
  <recv response="404" rtd="true" next="2" optional="true" />
<recv response="200" rtd="true" next="3" crlf="true" />

<label id="2" />
<nop>
  <action>
    <exec int_cmd="stop_call" />
  </action>
</nop>

```

```

<label id="3" />
<!-- Definition of the response time repartition table (unit is ms). -->
<ResponseTimeRepartition value="30, 50, 80, 100, 150, 200, 300, 400, 500, 600, 700, 1000, 1200,
1400, 1500" />

<!-- Definition of the call length repartition table (unit is ms). -->
<CallLengthRepartition value="30, 50, 80, 100, 150, 200, 300, 400, 500, 600, 700, 1000, 1200, 1400,
1500" />
</scenario>

```

Keywords from SIPp

"UDP" or "TCP"

```

<scenario name="UAC Basic Registration">
<send retrans="500" start_rtd="true">
  <![CDATA[

REGISTER sip:[field1] SIP/2.0
Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
From: [field0] <sip:[field0]@[field1]>;tag=[call_number]
To: [field0] <sip:[field0]@[field1]>
Call-ID: [call_id]
CSeq: [cseq] REGISTER
Contact: [field0] <sip:[field0]@[local_ip]:[local_port]>;expires=691200
Max-Forwards: 70
Content-Length: 0

  ]]>
</send>

```

Scenarios

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">

<!--          Sipp 'uac' scenario with pcap (rtp) play          -->
<!--                                     -->

```

```

<scenario name="UAC with media">
  <!-- In client mode (sipp placing calls), the Call-ID MUST be -->
  <!-- generated by sipp. To do so, use [call_id] keyword. -->
  <send retrans="500">
    <![CDATA[

      INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
      From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
      To: sut <sip:[service]@[remote_ip]:[remote_port]>
      Call-ID: [call_id]
      CSeq: 1 INVITE
      Contact: sip:sipp@[local_ip]:[local_port]
      Max-Forwards: 70
      Subject: Performance Test
      Content-Type: application/sdp
      Content-Length: [len]

      v=0
      o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
      s=-
      c=IN IP[local_ip_type] [local_ip]
      t=0 0
      m=audio [auto_media_port] RTP/AVP 8 101
      a=rtpmap:8 PCMA/8000
      a=rtpmap:101 telephone-event/8000
      a=fmtp:101 0-11,16

    ]]>
  </send>

  <recv response="100" optional="true">
  </recv>

  <recv response="180" optional="true">
  </recv>

  <!-- By adding rrs="true" (Record Route Sets), the route sets -->
  <!-- are saved and used for following messages sent. Useful to test -->

```

```

<!-- against stateful SIP proxies/B2BUAs. -->
<recv response="200" rtd="true" crlf="true">
</recv>

<!-- Packet lost can be simulated in any send/recv message by -->
<!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->
<send>
  <![CDATA[

    ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
    Call-ID: [call_id]
    CSeq: 1 ACK
    Contact: sip:sipp@[local_ip]:[local_port]
    Max-Forwards: 70
    Subject: Performance Test
    Content-Length: 0

  ]]>
</send>

<!-- Play a pre-recorded PCAP file (RTP stream) -->
<nop>
  <action>
    <exec play_pcap_audio="pcap/g711a.pcap"/>
  </action>
</nop>

<!-- Pause 8 seconds, which is approximately the duration of the -->
<!-- PCAP file -->
<pause milliseconds="8000"/>

<!-- Play an out of band DTMF '1' -->
<nop>
  <action>
    <exec play_pcap_audio="pcap/dtmf_2833_1.pcap"/>
  </action>

```

```

</nop>

<pause milliseconds="1000"/>

<!-- The 'crlf' option inserts a blank line in the statistics report. -->
<send retrans="500">
  <![CDATA[

    BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
    Call-ID: [call_id]
    CSeq: 2 BYE
    Contact: sip:sipp@[local_ip]:[local_port]
    Max-Forwards: 70
    Subject: Performance Test
    Content-Length: 0

  ]]>
</send>

<recv response="200" crlf="true">
</recv>

<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>

<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>

</scenario>
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">

<!--          Sipp 'uac' scenario with pcap (rtp) play          -->
<!--          -->

<scenario name="UAC with media">

```

```

<!-- In client mode (sipp placing calls), the Call-ID MUST be -->
<!-- generated by sipp. To do so, use [call_id] keyword. -->
<send retrans="500">
  <![CDATA[

    INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
    To: sut <sip:[service]@[remote_ip]:[remote_port]>
    Call-ID: [call_id]
    CSeq: 1 INVITE
    Contact: sip:sipp@[local_ip]:[local_port]
    Max-Forwards: 70
    Subject: Performance Test
    Content-Type: application/sdp
    Content-Length: [len]

    v=0
    o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
    s=-
    c=IN IP[local_ip_type] [local_ip]
    t=0 0
    m=audio [auto_media_port] RTP/AVP 8 101
    a=rtpmap:8 PCMA/8000
    a=rtpmap:101 telephone-event/8000
    a=fmtp:101 0-11,16
    m=video [media_port+2] RTP/AVP 34
    a=rtpmap:34 H263/9000
    a=fmtp:34 CTIF=1 QCIF=1 SQCIF=1

  ]]>
</send>

<recv response="100" optional="true">
</recv>

<recv response="180" optional="true">
</recv>

```

```

<recv response="183" optional="true">
</recv>

<!-- By adding rrs="true" (Record Route Sets), the route sets -->
<!-- are saved and used for following messages sent. Useful to test -->
<!-- against stateful SIP proxies/B2BUAs. -->
<recv response="200" rtd="true" crlf="true">
</recv>

<!-- Packet lost can be simulated in any send/recv message by -->
<!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->
<send>
  <![CDATA[

    ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
    Call-ID: [call_id]
    CSeq: 1 ACK
    Contact: sip:sipp@[local_ip]:[local_port]
    Max-Forwards: 70
    Subject: Performance Test
    Content-Length: 0

  ]]>
</send>

<!-- Play a pre-recorded PCAP file (RTP stream) -->
<nop>
  <action>
    <exec play_pcap_audio="pcap/g711a.pcap"/>
  </action>
</nop>

<nop>
  <action>
    <exec play_pcap_video="MMdemoMainmenuH263.pcap"/>
  </action>

```

```

</nop>

<!-- Pause 8 seconds, which is approximately the duration of the    -->
<!-- PCAP file                                                    -->
<pause milliseconds="8000"/>

<!-- Play an out of band DTMF '1'                                -->
<nop>
  <action>
    <exec play_pcap_audio="pcap/dtmf_2833_1.pcap"/>
  </action>
</nop>

<pause milliseconds="1000"/>

<!-- The 'crlf' option inserts a blank line in the statistics report. -->
<send retrans="500">
  <![CDATA[

    BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: sipp <sip:sipp@[local_ip]:[local_port]>;tag=[pid]SIPpTag09[call_number]
    To: sut <sip:[service]@[remote_ip]:[remote_port]>[peer_tag_param]
    Call-ID: [call_id]
    CSeq: 2 BYE
    Contact: sip:sipp@[local_ip]:[local_port]
    Max-Forwards: 70
    Subject: Performance Test
    Content-Length: 0

  ]]>
</send>

<recv response="200" crlf="true">
</recv>

<!-- definition of the response time repartition table (unit is ms) -->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>

```

```
<!-- definition of the call length repartition table (unit is ms) -->  
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>  
  
</scenario>
```





Appendix D

Performance Analysis of Asterisk SIP Server for Emergency Communications

Performance Analysis of Asterisk SIP Server for Emergency Communications

Sirinya Tabsombat¹, Mongkol Raksapatcharawong², Katsunori Yamaoka³, Teera Phatrapornnant⁴, Waranyoo Phiwthongkham⁴, and Phisanu Duangtanoo⁴

¹TAIST Tokyo Tech, ICTES Program, Department of Electrical Engineering, Kasetsart University, Bangkok, Thailand

²Department of Electrical Engineering, Kasetsart University, Bangkok, Thailand

³Department of Communications and Integrated Systems, Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo, Japan

⁴National Electronics and Computer Technology Center, Pathumthani, Thailand

tsirinya@hotmail.com, fengmkr@ku.ac.th, yamaoka.ss@titech.ac.jp, teera.phatrapornnant@nectec.or.th, waranyoo.phiwthongkham@nectec.or.th, phisanu.duangtanoo@nectec.or.th

Abstract— A policy-based network such as an interoperability system for emergency situation needs a solid reliability. The analysis of the performance characteristic can lead to learning technique to optimize its policy decisions. This paper describes the results of experimental testbed scenarios and the laboratory test of an interoperability system using asterisk SIP server running under Linux OS to evaluate how real-time applications (such as voice and video conferencing) may suffer under a “critical situation mission”. Using simulation, in this paper we can infer minimum configuration requirement for a possible capabilities such as the possible maximum calls, the appropriate call setup time and the acceptable voice quality for emergency communications.

Keywords– Voice Over IP, SIP, Asterisk, SIPP, IP-PBX

I. INTRODUCTION

Crisis and Emergency Communications fundamentally differ from day-to-day communications activities in many ways, including with: need for immediate response to the callers, need for appropriate task management, and need for a flexible communications coordination.

For the purpose of this system, “Callers” means police, firefighter and ambulance services and any

other emergency public services which are ultimately responsible for dispatching emergency services.

As a backup communication system for an emergency situation, the constraint of emergency situation network service includes a traffic management for a large amount of calls and a capability in maintaining a quality of communication simultaneously.

In this paper, we present a performance analysis of Asterisk-based IP-PBX to determine the reliability in emergency communication mission. The important issue is a relationship between a service efficiency and a voice quality. There are few disadvantages of voice over IP [1]. The main complaint regarding VoIP has a lower level of voice quality than regular telephone technology. Therefore VoIP requires a large amount of data to be compressed and transmitted, then uncompressed and delivered, all in a relatively small amount of time. The policy in developing this VoIP interoperability system expects the system has immediate response. The callers require good voice quality such as low echo, delay or packet loss. Where system's requirements are:

- 1: The system can support a number of incoming calls sufficient to handle a critical situation.
- 2: The system must be readily deployed in response to critical situation.
- 3: The system performance would not be severely degraded by a large amount of packet loss.

II. SYSTEM ARCHITECTURE

This IP-PBX system uses a single server for each node. The system organized by Linux platform handle with Asterisk PBX and open source applications, providing both voice and video applications support.

A. Session Initiation Protocol:

The Session Initiation Protocol (SIP) is a text-based protocol that was originally designed by Henning Schulzrinne and Mark Handley starting in 1996. The SIP protocol is a TCP/IP-based Application Layer protocol, with syntax similar to that of HTTP. SIP typically uses port number 5060 or 5061 both for UDP and TCP, and it may implement other transport protocols. A motivating goal for SIP was to provide a signaling and call setup protocol for IP-based communications. SIP also defines server network elements. It has been standardized and governed primarily by the IETF, while other protocols, such as H.323, have been traditionally associated with the International Telecommunication Union (ITU).

Two SIP endpoints can communicate without any intervening SIP infrastructure, because there are various types of gateways compatible at the edge between a SIP network and other networks which is why the protocol is described as peer-to-peer. This approach is often impractical for a typical public service telephone network, therefore the advantages of simple, expansibility and modularization can provides the possibility of interoperability communication system implementation.



Figure 1. The communication between SIP Servers



Figure 2. The interoperability communication system on each node performed by Asterisk PBX on the SIP Server

B. Hardware Architecture

The prototype configuration for VoIP server section consisted of the following details:

The Intel® Server SR1530SH ,Processor : 64-bit Multi-Core Intel® Xeon® 3200/3000 sequence 1.33 GHz, 8 GB RAM, and one Gigabit Ethernet Switch Cisco SRW224G4P.

A SIP phone is a hardware-based or software-based SIP user agent. The hardware phones provided in this system is Cisco Linksys SPA932.

C. Software Architecture

Asterisk : An open source software and is available free to all under the GNU General Public License. It includes many features available in proprietary PBX systems, supporting a mix of traditional and VoIP telephony services.

The following software for server node running under Linux Cent OS 5.0 kernel 2.6.18-164 were deployed:

Asterisk version 1.6.2.0, FreePBX version 2.5.1.0

The SIP software phone for client node is X-Lite version 3.0. X-Lite is developed by CounterPath, available on <http://www.counterpath.com>.

III. PERFORMANCE EVALUATION

The performance test was set to monitoring the characteristics of the system on both service quality and voice quality.

Service quality: can be determined by the reliability on traffic management such as the call setup time, the call setup failure, disconnect probability etc.

Voice quality: can be analyzed from the transport signals such as SIP messages timestamps, latency, jitter and packet loss.

A. Test Methodology

The test used the SIPp 3.1, a free open software test tool / traffic generator for the SIP protocol. The SIPp is licensed under the GNU General Public License. It includes a few basic of Benchmarking SIP Server performance using for emulating user agents calling a SIP server [12]. Reaching high call rates and/or high number of simultaneous SIP calls is possible.

1) SIP calls monitoring:

The traffic load is distributed across the SIP server using Cisco SRW224G4P Gigabit Switch. The SIPp is installed on a client computer. The characteristic of average response time and call retractions can be monitored by the SIP client computer.



Figure 3. Experiment for system's response

2) Traffic Signal Monitoring:

The transmission/receiving packets from each SIP call includes payload type identification, sequence numbering, and timestamping can be monitored and recorded by Wireshark software tool Version 0.99.7 (SVN Rev 23910).

Wireshark is a network protocol analyzer for Unix and Windows, it available under the GNU General Public License. With packet filtering, we considered only the SIP RTP protocol packets.

In this experiment, a pair set of VoIP call was created to try a connection and monitor a voice/media quality while the traffic were generated.

Therefore, the other suit of SIP client computer was provided. The X-Lite softphone was installed on this computer including with Wireshark. The VoIP phone was set for the other side of the connection.

In media transmission, we replaced the VoIP phone with X-Lite softphone on the other one client computer.

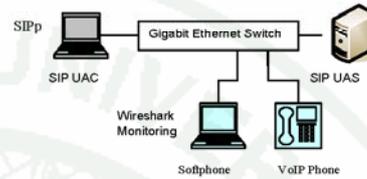


Figure 4. Experiment for voice quality

B. SIP Traffic Model

The communications between SIP client and SIP server use both UDP protocol. And a stream of the Real-Time Transport Protocol (RTP) packets has been generated and transported to the traffic simulations.

RTP runs on top of transport protocols, typically UDP, and provides real-time applications with end-to-end delivery services such as payload type identification and delivery monitoring. RTP provides transport of data contains with a message sequence number to allow applications to detect packet loss, packet duplication, or packet reordering.

By using an XML pattern for a SIP Call Flow, both voice and video traffic signals were generated. To set up an voice transmission session, The simulation was assigned to use G.711 codec. And in video transmission, H.264 video supported codec was used. G.711 - is the widely used PCM standard, encoding 8 bit speech samples at 8000 samples per second, giving 64 kilobits per second of digital speech data.

Some characteristics of G.711 codec [2] are shown in Table I.

TABLE I. SOME CHARACTERISTICS OF G.711 CODEC

Standard	Bandwidth	Sample period	Ethernet Bandwidth
ITU-T G.711	64 kbps	20 ms	95.2 kbps

The bitrate supported for H.264 codec according to RFC 3550 [3] RTP payload format depends on the sampling rate and number of channels selected. Each manufacturer customizes its H.264 profile to suit its needs, which means no two H.264 profiles are necessarily created equal [4]. For the worst case scenarios we used variable bit rate (VBR) where the bandwidth will increase dramatically to attempt to achieve this purpose.

C. Call Scenarios

SIP messages are often transferred by means of transport protocol such as UDP. A session invitation consists of one INVITE request from User Agent Client (UAC) to User Agent Server (UAS). The server sends immediately a "100 Trying" reply to stop retransmissions and forwards the request further. Then, all provisional responses are sent back to the caller (UAC) such as "180 Ringing" response, therefore the phone starts ringing. The "2xx" is generated final responses to INVITE, ACK is the indicates reception of the 2xx by the UAC. The RTP transmission is generated when the call set up succeeded [5].

On the other hand, the UAC would send a retransmission message to request again if receives no reply within a period of time. The SIP Call Flow in. Figure 3. describes the method of call simulation scenarios, where the SIP messages are transferred directly by User Agents without proxy interface.

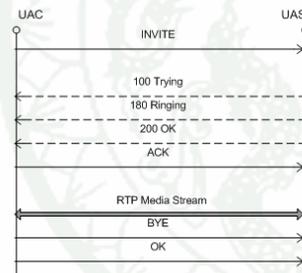


Figure 5. SIP Call Flow

To perform a fairly complete set of SIP signaling, SIP transaction state machines are described below. These transaction states are not a complete SIP stack but provides a limited set of features common to any kind of SIP Agents. The state machines are provided as

drawings within the SIP RFC3261.txt (section 17.1 and 17.2) [5].

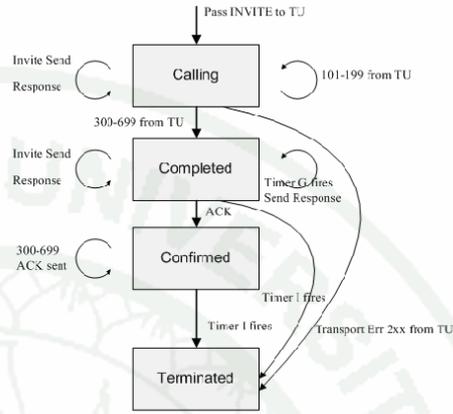


Figure 6. INVITE Server Transaction

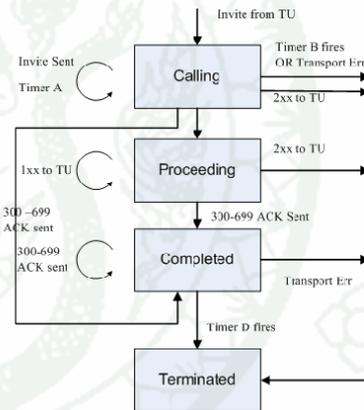


Figure 7. INVITE Client Transaction

Where TU stands for Transaction User. The "Completed" state exists to buffer any additional response retransmissions that may be received. The

"Confirmed" state is to absorb any additional ACK messages that arrive.

In SIP traffic simulation, we separated a range of call rate by 50, 100, 150 and 200 calls per second. There are the length of call period from call setup time till call terminated. The appropriate period for our dial plan configured [12] is about 10 seconds. In our observation, the amount of concurrent calls are the combination of each current generated calls and a number of calls from previous seconds which still alive. For example at 50 call/s rate, after 5 seconds, the number of concurrent calls are 250 calls. The value is much exceed and difficult to analyze the real behaviors of 50 simultaneous calls. By limiting the concurrent calls, we found that no more calls were generated while the concurrent calls are reach the call rate that was set, so we can determine the exactly behaviors of each simultaneous call rate. Therefore, when running the load test of 50 calls per second, the 50 simultaneous calls were arrived simultaneously at the first second and then no number of calls were generated until the next 10 seconds after the previous calls were terminated. The test period were run for 60 seconds each case, therefore after 60 seconds, the totally call messages are 300 instead of 3000. We compared the test case with more length of period and found that there is no effect to the test results.

The simulations were perform more than 5 times each case, and the captures selected for analysis represent the average values of all simulations.

The test results will be displayed the total number of incoming calls, retransmission messages and error messages. The best condition is all calls completed without retransmission and without any unexpected message. A low percentage of retransmission may be acceptable for very heavy traffic conditions.

IV. SIMULATION RESULTS

A. Voice traffic

The following screenshot on this page demonstrates the SIPp performance test with active voice traffic generated for a call rate of 50 calls per second, with 10 seconds of call duration.

Message	Messages	Retrans	Timeout	Unexpected-Msg
INVITE	350	0	0	0
ACK	350	0	0	0
BYE	300	0	0	0

Figure 8. Voice traffic simulation at 50 call/s

From the simulation screen capture, the result can be analyzed that 100 % of the calls are completed on the first try.

We found that there are no retransmission, timeout or any unexpected message which would indicate the failed calls occurred. We can determine from the test result that the system can completely performance at this rate of simultaneous call.

Response Time	Repartition
10 ms <= m <=	350
20 ms <= m <=	0
30 ms <= m <=	0
40 ms <= m <=	0
50 ms <= m <=	0
1000 ms <= m <=	0
1500 ms <= m <=	0
2000 ms <= m <=	0

Call Length	Repartition
10 ms <= m <=	300
20 ms <= m <=	0
30 ms <= m <=	0
40 ms <= m <=	0
50 ms <= m <=	0
1000 ms <= m <=	0
1500 ms <= m <=	0
2000 ms <= m <=	0

Figure 9. Voice traffic simulation at 50 call/s

The average response time repartition inform us the average call setup time that started from the INVITE message until the ACK message has been sent. In the call rate of 50 call/second, the total calls are 300 messages and the call setup time can be determine that 100% of the calls are completed within <10 ms.

The average call length repartition indicates the ability in call clear management. Due to the simulation condition, 10 seconds of call duration was assigned to the system. From the result, after 350 invite messages were generated, the 300 messages at the completed state were terminated. The call event of all 300 messages were cleared within 10 seconds, we can determine that we can determine that the system can handled the call clear management properly at the call rate of 50 call/s.

Another case of simulation provided the result as shown in Table V and Table VI.

In determine the performance quality of the system, the ETSI EG201 769 standard [8] for VoIP Provider is used in compare with the test results.

TABLE II. THE FIRST EXPERIMENT OF VOICE TRAFFIC SIMULATION

Parameter	Unit	Call Rate (call/sec)			
		50	100	150	200
Retrans.	%	0	0	0	>50
Failed Call	%	0	0	0	>90

TABLE III. THE FIRST EXPERIMENT VOICE TRAFFIC SIMULATION

Parameter	Unit	Stand ard*	Call Rate (call/sec)			
			50	100	150	200
Call set-up time	< sec	5.88	0.01	0.03	0.04	>0.2
Call setup error	<%	0.005	0	0	0	>90
Call setup failure	<%	18.00	0	0	0	>90
Disconnect Probability	<%	0.003	0	0	0	0.25
Call clear failure	<%	0.05	0	0	0	0

*ETSI EG201 769 standard for VoIP Provider

After the simulation, we found that there are no retransmission and error messages at 150 call/s and the system can work properly until the test terminated.

At the call rate of 200 call/s, a high number of retransmission increased after few seconds and the traffic then stopped. The system had no response to the incoming calls and the simulation was forced to be terminated. We can determine that the call rate is over limited. We tried again to met the limit of simultaneous call by increasing the call rate between 150 call/s to 200 call/s. So the call rate were separated by 160, 170, 180 and 190 call/s.

The simulation result is provided in Table VII and Table VIII.

TABLE IV. THE SECOND EXPERIMENT OF VOICE TRAFFIC SIMULATION

Parameter	Unit	Call Rate (call/sec)			
		160	170	180	190
Retrans.	%	0	>50	>50	>50
Failed Call	%	0	>90	>90	>90

TABLE V. THE SECOND EXPERIMENT OF VOICE TRAFFIC SIMULATION

Parameter	Unit	Standard*	Call Rate (call/sec)			
			160	170	180	190
Call set-up time	< sec	5.88	0.04	>0.2	>0.2	>0.2
Call setup error	<%	0.005	0	>90	>90	>90
Call setup failure	<%	18.00	0	>90	>90	>90
Disconnect Probability	<%	0.003	0	0	0	0
Call clear failure	<%	0.05	0	0	0	0.06

*ETSI EG201 769 standard for VoIP Provider

At 160 call/s there are still no retransmission and error messages. The system can work properly until the test terminated. But at the call rate of 170 call/s a high percent of retransmission messages, timeout messages and unexpected messages were given.

The system can work properly before 30 seconds of simulation, and then becomes error after that, therefore, the UAS has little response to UAC while the retransmission increased highly.

B. Video traffic

For Video traffic monitoring, the same scenarios and simulations were run with RTP media packets.

Figure 8. gives the response of the call rate at 150 call/s, some retransmission and error messages occurred at this call rate.

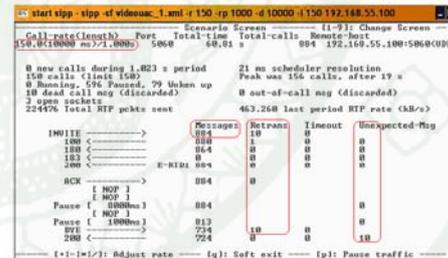


Figure 10. video traffic simulation at 150 call/s

At 150 call/s, the result can be analyzed as details:

- 97.73 % of the calls are completed on the first try.
- 0.45 % of the calls fail on the first attempt and complete on the second attempt.
- 1.81% of the calls fail on the first two attempts and complete on the third attempt.
- 0% of the calls fail on the first three attempts and complete on the fourth attempt.

The total call retransmission messages are 2.37 percents, and the total call fail are 1.13 percents. The system can work properly until the test terminated.

Fro Figure 11. The maximum average response time is over 200 ms with few number of calls, and 34.51 percents of call clear time are exceed 10 seconds.

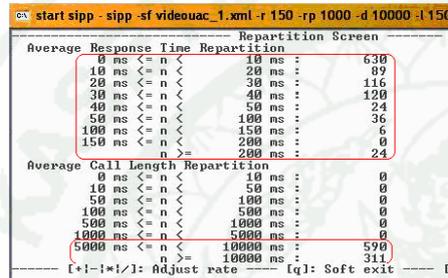


Figure 11. video traffic simulation at 150 call/s

The simulation result has shown in Table IX and Table X.

TABLE VI. THE FIRST EXPERIMENT OF VIDEO TRAFFIC SIMULATION

Parameter	Unit	Call Rate (call/sec)			
		50	100	150	200
Retrans.	%	0	0	10	>50
Failed Call	%	0	0	2.34	> 90

TABLE VII. THE FIRST EXPERIMENT OF VIDEO TRAFFIC SIMULATION

Parameter	Unit	Stand ard*	Call Rate (call/sec)			
			50	100	150	200
Call set-up time	< sec	5.88	0.01	0.03	0.04	>0.2
Call setup error	<%	0.005	0	0	0	>90

Call setup failure	<%	18.00	0	0	0	>90
Disconnect Probability	<%	0.003	0	0	0	0
Call clear failure	<%	0.05	0	0	0	0

*ETSI EG201 769 standard for VoIP Provider

After simulations, the call rate has over limit at 200 call/s with a very high percent of error messages, and the traffic was then stopped. We tried again by separated the range of call rate in 160, 170, 180 and 190 call/s. The simulation result is provided in Table XI and Table XII.

TABLE VIII. THE SECOND EXPERIMENT OF VIDEO TRAFFIC SIMULATION

Parameter	Unit	Call Rate (call/sec)			
		160	170	180	190
Retrans.	%	12	> 50	> 50	> 50
Failed Call	%	>90	> 90	>90	>90

TABLE IX. THE SECOND EXPERIMENT OF VIDEO TRAFFIC SIMULATION

Parameter	Unit	Stand ard *	Call Rate (call/sec)			
			160	170	180	190
Call set-up time	< sec	5.88	0.03	>0.2	>0.2	>0.2
Call setup error	<%	0.005	>90	>90	>90	>90
Call setup failure	<%	18.00	>90	>90	>90	>90
Disconnect Probability	<%	0.003	0	0	0	0
Call clear failure	<%	0.05	0	0	0	0

*ETSI EG201 769 standard for VoIP Provider

From the second experiment results, the system can work properly until 150 call/s, after run at higher call rate, the retransmission and error messages increased very fast, and the test then were forced terminated.

- 97.73 % of the calls are completed on the first try.
- 0.45 % of the calls fail on the first attempt and complete on the second attempt.
- 1.81% of the calls fail on the first two attempts and complete on the third attempt.
- 0% of the calls fail on the first three attempts and complete on the fourth attempt.

The total call retransmission messages are 2.37 percents, and the total call fail are 1.13 percents. The system can work properly until the test terminated.

Fro Figure 11. The maximum average response time is over 200 ms with few number of calls, and 34.51 percents of call clear time are exceed 10 seconds.

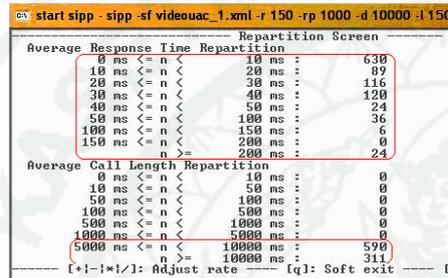


Figure 11. video traffic simulation at 150 call/s

The simulation result has shown in Table IX and Table X.

TABLE VI. THE FIRST EXPERIMENT OF VIDEO TRAFFIC SIMULATION

Parameter	Unit	Call Rate (call/sec)			
		50	100	150	200
Retrans.	%	0	0	10	>50
Failed Call	%	0	0	2.34	> 90

TABLE VII. THE FIRST EXPERIMENT OF VIDEO TRAFFIC SIMULATION

Parameter	Unit	Stand ard*	Call Rate (call/sec)			
			50	100	150	200
Call set-up time	< sec	5.88	0.01	0.03	0.04	>0.2
Call setup error	<%	0.005	0	0	0	>90

Call setup failure	<%	18.00	0	0	0	>90
Disconnect Probability	<%	0.003	0	0	0	0
Call clear failure	<%	0.05	0	0	0	0

*ETSI EG201 769 standard for VoIP Provider

After simulations, the call rate has over limit at 200 call/s with a very high percent of error messages, and the traffic was then stopped. We tried again by separated the range of call rate in 160, 170, 180 and 190 call/s. The simulation result is provided in Table XI and Table XII.

TABLE VIII. THE SECOND EXPERIMENT OF VIDEO TRAFFIC SIMULATION

Parameter	Unit	Call Rate (call/sec)			
		160	170	180	190
Retrans.	%	12	> 50	> 50	> 50
Failed Call	%	>90	> 90	>90	>90

TABLE IX. THE SECOND EXPERIMENT OF VIDEO TRAFFIC SIMULATION

Parameter	Unit	Stand ard *	Call Rate (call/sec)			
			160	170	180	190
Call set-up time	< sec	5.88	0.03	>0.2	>0.2	>0.2
Call setup error	<%	0.005	>90	>90	>90	>90
Call setup failure	<%	18.00	>90	>90	>90	>90
Disconnect Probability	<%	0.003	0	0	0	0
Call clear failure	<%	0.05	0	0	0	0

*ETSI EG201 769 standard for VoIP Provider

From the second experiment results, the system can work properly until 150 call/s, after run at higher call rate, the retransmission and error messages increased very fast, and the test then were forced terminated.

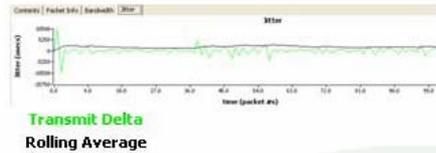


Figure 16. Average Jitter while perform the video traffic simulation at 150 call/s

In Figure 15, average jitter is 87.63 ms, and there are 2 dropped packets.

From the experiments, the average packet loss measured from Wireshark both from voice and video traffic are less than 1 second.

The number of jitter, packet dropped and packet loss are not affect the voice quality. These values represent the averages values that can be measured. There are many wrong order packets occurred both in voice and video traffic signal. For improving the voice quality, we need to control the packet loss and packet delay.

VI. CONCLUSION

In this paper we described a model of the SIP-based agent server in the implementation for Emergency Communication. We also presented testing scenarios for provisional response in SIP user agent. The test results perform the voice quality evaluation for internal system of each server node, without any trunk or network link. In implementing to the real application we should considered the voice codec and another environments, such as the transcoding [10], silence suppression, and the fact that the jitter, delay or packet loss will occur in both forward and reverse directions. On the other hand, the tests are performed under the minimum configuration therefore it has no QoS or dejitter buffer assignments. The appropriate configurations of QoS [11], dejitter buffer and bandwidth management will lead the better performance in both traffic management and voice quality.

VII. FUTURE WORK

The QoS management for the traffic load will be implemented to the system in order to increase the number of possible simultaneous calls. We also plan to find the solutions for the packet loss control to ensure the voice quality those two significant issues to meet the optimum performance.

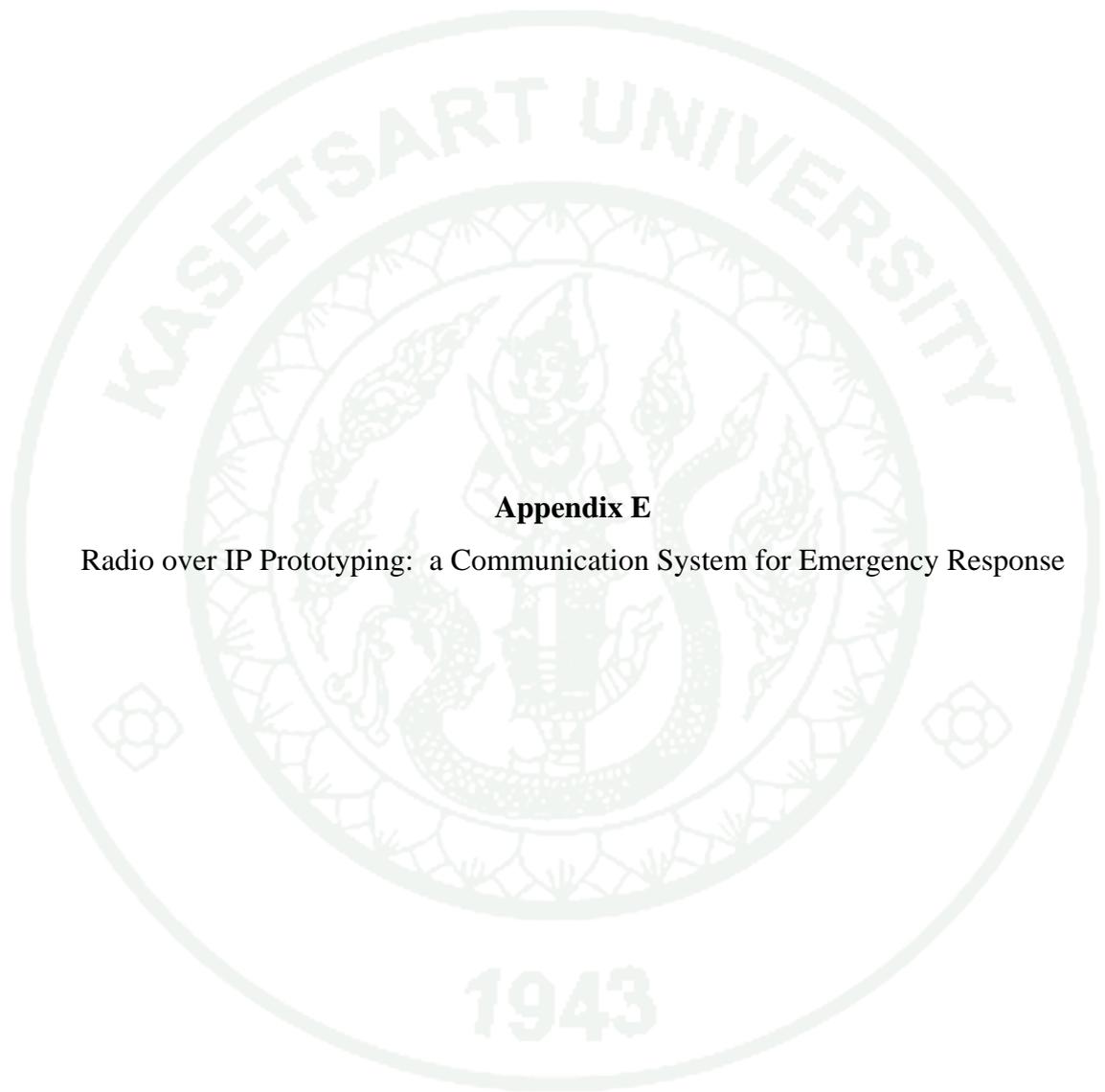
ACKNOWLEDGMENT

This research is financially supported by Thailand Advanced Institute of Science and Technology - Tokyo Institute of Technology (TAIST-Tokyo Tech), National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology (Tokyo Tech) and Kasetsart University (KU). The authors would like to express their thanks to MICT (Ministry of Information and Communication Technology) for supporting this research through a consultancy project entitled "Prototyping a Communication System for Emergency Response via IP", grant no. 27/2552.

REFERENCES

- [1] Christopher M. White, Josh Raymond, and Keith A. Teague. "A Real-Time Network Simulation Application for Multimedia over IP", 2245- 2249. Proc. IEEE. Signals, Systems and Computers, Conference Record of the Thirty-Eighth Asilomar, November 2004.
- [2] Recommendation G.711 : Pulse code modulation (PCM) of voice frequencies, November 2004.
- [3] ITU-T H.264 SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services – Coding of moving video, March 2005.
- [4] H.264 Considerations : Nine Key Considerations When Selecting a Video Compression Algorithm. www.iqeye.com/iqeye/H.264_Considerations.pdf
- [5] YuanZhang. "SIP based over IP network and its interworking with the PSTN", Electronics & Communication Engineering Journalpp. 273-282, December 2002
- [6] Mohiuddin Ahmed, Abdul Malik Mansor "CPU dimensioning on performance of Asterisk VoIP

- PBX", Proceedings of the 11th communications and networking simulation symposium, 2008.
- [7] Standards Track RFC 3261 SIP: Session Initiation Protocol, <http://www.ietf.org/rfc/rfc3261.txt>, June 2002.
- [8] ETSI E.G. 201 769 V1.1.2 Speech Processing, Transmission & Quality Aspects (STQ), October 2000.
- [9] Abdelbasset Trad, Farukh Munir, Thierry Turletti and Hossam A. 2006. VoIP Capacity Evaluation in IEEE 802.11e WLAN Environment, 828- 832. Proceedings of IEEE. 3rd Consumer Communications and Networking Conference.
- [10] Ravi Shankar Ramakrishnan and P. Vinod kumar Abdelbasset, "Performance Analysis of Different Codecs in VoIP Using SIP", <http://tifac.velammal.org/CoMPC/articles/30.pdf>, 2009.
- [11] Rozalina Dimova, Georgi Georgiev, Zlatko Stanchev, "Performance Analysis of QoS Parameters for Voice over IP Applications in a LAN Segment", <http://csconf.org/Volume1/page232.pdf>, International Scientific Conference Computer Science 2008.
- [12] Razvan Rughinis and Cristian Iconaru, "A Practical Analysis of Asterisk SIP Server Performance", <http://conference.cluj.roedu.net/papers/25.pdf>, 2009.



Appendix E

Radio over IP Prototyping: a Communication System for Emergency Response

Radio over IP Prototyping: a Communication System for Emergency Response

S. Tabsombat¹, N. Pimpuch², A. Hiranya-ekaparb², M. Raksapatcharawong², K. Yamaoka³, T. Phatrapornnant⁴, W. Phiwthongkham⁴, and P. Duangtanoo⁴

¹TAIST Tokyo Tech, ICTES Program, Department of Electrical Engineering, Kasetsart University, Bangkok, Thailand (tsirinya@hotmail.com)

²Department of Electrical Engineering, Kasetsart University, Bangkok, Thailand (fengmkr@ku.ac.th)

³Department of Communications and Integrated Systems, Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo, Japan (yamaoka.ss@titech.ac.jp)

⁴National Electronics and Computer Technology Center, Pathumthani, Thailand ({teera.phatrapornnant, waranyoo.phiwthongkham, phisanu.duangtanoo}@nectec.or.th)

ABSTRACT

In this paper, we present a Linux-based Radio over IP system. The system is a cost-effective IP interoperability communication system that integrated the mobile radio, typical public switched telephone network lines, cell phone, native IP devices etc. All equipment can access the communication with each other via IP network using hardware-software coordination on asterisk SIP IP-PBX running under Linux OS. After the evaluation of performance tests, we can determine the basic configurations for the system in order to readily for use as an interoperability system to support an emergency situation communication.

Keywords: Radio Over IP, SIP, Asterisk, IP-PBX, interoperability system

1. INTRODUCTION

In this approach, we develop one model of an IP Interoperability and Collaboration Suite (IPICS)[1][2][3][4][5] that can integrate the mobile radio systems, typical public switched telephone network lines, cell phone, native IP devices such as IP telephone systems, Softphone applications and all of these devices in order to communicate with each other, using hardware-software coordination.

Radio interoperability system remains a challenge[2]. It still needs more improving of the quality such as voice quality, data quality etc. or even think about the possibility to deal with other typical communication devices.

The aim of developing RoIP convergence communication system is to provide a tool that enables emergency response agencies to exchange critical communications and data with one another, permitting them to work together effectively. In general, when the typical communication systems is down, the mobile radio is normally used. Communication via Radio over IP system allows various mobile radio systems—such as HF, UHF, VHF, and citizen channel to communicate with each other during emergency situations. The more efficiency of the system is the typical communication device such as telephone, cell phone and VoIP device are allowed to access to radio communication, via IP.

The more challenge in developing RoIP system is the device's properties, the mobile radio uses a half-duplex communication, while telephone uses the full duplex. The interconnection between radio and telephone then more complicated than general VoIP communication.

In RoIP system, PTT / COR and voice signals are sent across the LAN / WAN via RoIP gateway in IP format.

The Gateway connects to a server using SIP and RTP/IP protocols.

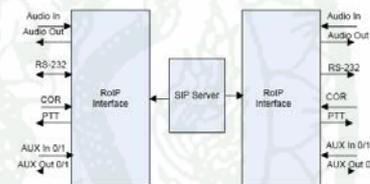


Figure 1. Block diagram of Radio over IP system

2. THE SIP-BASED ROIP SYSTEM

The system is an IP-based communication using SIP, a TCP/IP-based Application Layer protocol, with syntax similar to that of HTTP. SIP typically uses port number 5060 or 5061 both for UDP and TCP, and it may implement other transport protocols. The IP-PBX software that performance as a core of the system is Asterisk version 1.6.2.0, running under Linux Cent OS 5.0 kernel 2.6.18-164.[6].

For the software development design, the most important for application is the system configuration method. The best operation can be performed under the appropriate cooperation between dial plan, protocol, codec[7], QoS management[8], etc.

In hardware development of this system, the performance of Radio over IP needs the Telephone-Radio controller circuit. The circuit including with 5 parts :

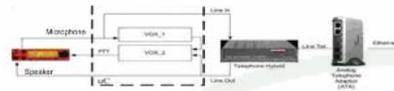


Figure 2. Block diagram of Telephone to Radio controller circuit

2.1 Telephone Hybrid

The circuit is designed to transform the normal telephone line and separate to Line in and Line out form. Therefore, the communication between telephones and radios can be enabled.

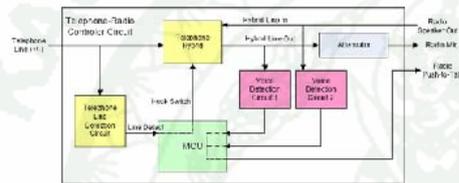


Figure 3. Radio-Phone cooperation via Telephone Hybrid Circuit

2.2 Telephone Line Detection

The circuit is designed to detect the telephone line based audio signal and transform to the MCU-readable signal. The MCU then can detect the Ring signal and switch the phone state to the off-hook or on-hook state.

2.3 Voice Detection

The circuit incorporates the necessary amplifiers, attenuators, level detectors, and control algorithm to form a high quality of VOX/PTT operation[5][9].

2.4 Attenuator

This device is adopted to reduces the amplitude or power of a signal from the Hybrid telephone circuit which the Line Out signal is too high to feed forward to microphone.

2.5 MCU

This unit is the controller for all process including with signal detection, timer, and other cooperation. The MCU is connected to 2 necessary systems:

- 1) Telephone system: to detect the line state in order to hook and hang up control.

- 2) Radio system: to detect the voice level for a Push-To-Talk control.

3. EMERGENCY COMMUNICATION NETWORK

We can separate the connection of this RoIP system link to two sections: the internal link and the external link. The internal link including with the communication between each component in one base such as wireless LAN and another devices. In extending the communication range, the Long-Range Wi-Fi can support the telecommunication to 30 km range of each base.

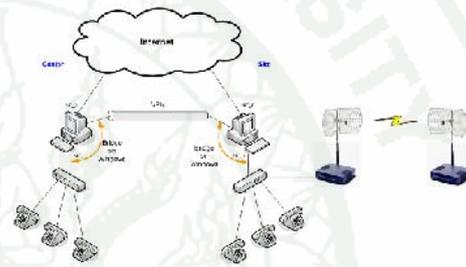


Figure 4. The diagram of the connection between two bases network

In setting up the Long-Range Wi-Fi, the software aid calculation and Antenna tracker are necessary in emergency situation. All significant parameters such as TX, RX gain, antenna gain etc. related to the distance setting need to be determined in fast an accurate. In our experiment, the EnGenius ECB-3610s wireless access point, the grid antenna and its equipment was used. The distance between the pair of Long-Range Wi-Fi is 800 m and the internet bandwidth is 54 Mbps. In this case of short range distance, the transmission power was need to be reduced [10].

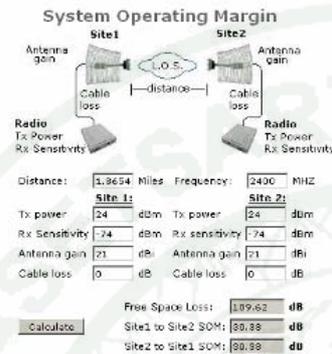


Figure 5. The example of using software in calculating the set up distance of Long-Range Wi-Fi

The external link is designed for the communication between each base. The connection between RoIP networks can be deployed to the private internet network concepts. In the real situation, only the authorized person can access to the network for security policy. Therefore, the concept of private network is implemented to our system. Some useful concept are VPN: Virtual Private Network and DDNS: Dynamic Domain Name Server. Both concepts can support the best performance to the system.

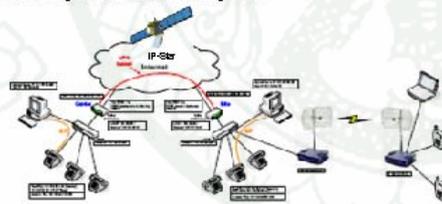


Figure 6. The diagram of network connection via satellite

4. PERFORMANCE EVALUATION

The communication via 2 Mbps IPstar broadband satellite link has been test for the distance of 900 km between network to network (from Chiang Mai province in the north of Thailand to Bangkok). In this performance test, all communication devices such as VoIP Phone, PSTN phone, Softphone and mobile radio had been conference together. The Long-Range Wi-Fi was used in order to extend the internet for video transmission to 3 km from the site. The media transmission was smoothly for normal motion, and the

voice communication gives a good quality in both radio, and phones.



Figure 7. The field test of the RoIP interoperability system interconnection between two network bases via satellite : during the virtual disaster situation in the National Crisis Management Exercise 2009(C-MEX 09).

From the field test via 2 Mbps IPstar broadband satellite link, the Bandwidth usage for voice transmission using G.711 codec is shown on table 1, and video transmission using h.264 codec is shown on table 2 .

Table 1. Voice Transmission via 2 Mbps IPstar Broadband satellite link using G.711 codec

Number of Call(s)	CPUs	Bandwidth (Receive/Transmit) KB/s
1	1%	23.02 / 23.12
2	1%	34.66 / 46.30
3	1%	59.45 / 61.64
4	1%	89.40 / 83.03
5	1%	107.73 / 105.10
6	1%	126.62 / 125.42
7	2%	144.84 / 145.76
8	2%	163.32 / 166.53
9	2%	181.46 / 182.51
10	3%	195.45 / 196.20

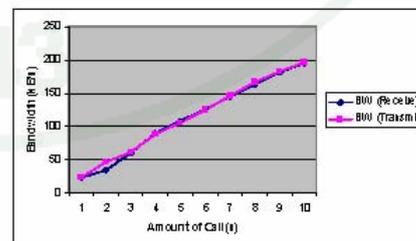


Figure 8. The statistics graph from the voice transmission via 2 Mbps IPstar Broadband satellite link using G.711 codec and SIP Trunk

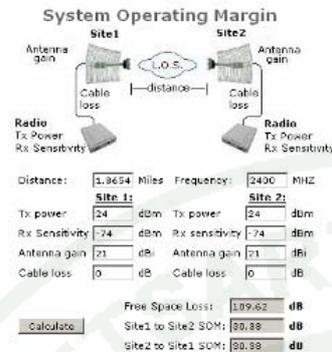


Figure 5. The example of using software in calculating the set up distance of Long-Range Wi-Fi

The external link is designed for the communication between each base. The connection between RoIP networks can be deployed to the private internet network concepts. In the real situation, only the authorized person can access to the network for security policy. Therefore, the concept of private network is implemented to our system. Some useful concept are VPN: Virtual Private Network and DDNS: Dynamic Domain Name Server. Both concepts can support the best performance to the system.



Figure 6. The diagram of network connection via satellite

4. PERFORMANCE EVALUATION

The communication via 2 Mbps IPstar broadband satellite link has been test for the distance of 900 km between network to network (from Chiang Mai province in the north of Thailand to Bangkok). In this performance test, all communication devices such as VoIP Phone, PSTN phone, Softphone and mobile radio had been conference together. The Long-Range Wi-Fi was used in order to extend the internet for video transmission to 3 km from the site. The media transmission was smoothly for normal motion, and the

voice communication gives a good quality in both radio, and phones.



Figure 7. The field test of the RoIP interoperability system interconnection between two network bases via satellite : during the virtual disaster situation in the National Crisis Management Exercise 2009(C-MEX 09).

From the field test via 2 Mbps IPstar broadband satellite link, the Bandwidth usage for voice transmission using G.711 codec is shown on table 1, and video transmission using h.264 codec is shown on table 2 .

Table 1. Voice Transmission via 2 Mbps IPstar Broadband satellite link using G.711 codec

Number of Call(s)	CPUs	Bandwidth (Receive/Transmit) KB/s
1	1%	23.02 / 23.12
2	1%	34.66 / 46.30
3	1%	59.45 / 61.64
4	1%	89.40 / 88.03
5	1%	107.73 / 105.10
6	1%	126.62 / 125.42
7	2%	144.84 / 145.76
8	2%	163.32 / 166.53
9	2%	181.46 / 182.51
10	3%	195.45 / 196.20

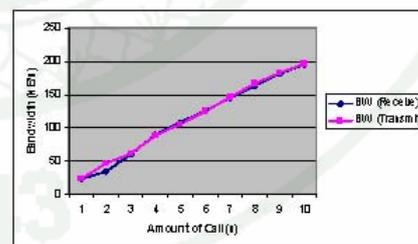


Figure 8. The statistics graph from the voice transmission via 2 Mbps IPstar Broadband satellite link using G.711 codec and SIP Trunk

The voice transmission via IP-Star using G.711 codec has a bandwidth consumption about 20 kB/s(Transmit) per 1 call. These values are the total values that included both the components and network bandwidth consumption. To determine the appropriate bandwidth management, we considered only the network bandwidth consumption that used a half of the total values. So, the bandwidth consumption per 1 call is about 10 KB/s for G.711 codec. From the experiment, when the concurrent calls reaches 10 calls, the bandwidth consumption is about 98 kB/s(196.20 /2) or 0.78 Mb/s. We can determine that the voice transmission via 2 Mbps (2 downlink/1 uplink) IP-Star broadband using G.711 codec can support up to 13 calls (the bandwidth consumption of 13 calls is about 1014 kb/s or 1 Mbps).

Table 2. Voice Transmission via 2 Mbps IPstar Broadband satellite link usingG.729 codec

Number of Call(s)	CPU%	Bandwidth (Receive/Transmit) KB/s
1	1%	14.63 / 14.38
2	1%	29.05 / 28.99
3	1%	40.76 / 43.33
4	1%	47.01 / 51.28
5	1%	62.37 /66.27
6	1%	73.43 /78.12
7	2%	88.43 /89.26
8	2%	97.34 /98.25
9	2%	109.03 /114.32
10	3%	122.43 / 125.27

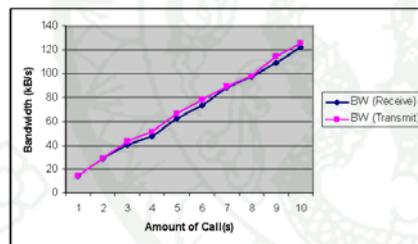


Figure 9. The statistics graph from the voice transmission via 2 Mbps IPstar Broadband satellite link using G.729 codec and SIP Trunk

From the experiment, the voice transmission via IP-Star using G.729 codec has a bandwidth consumption about 15 kB/s(Transmit) per 1 call. These values are the total values that included both the components and network bandwidth consumption. To determine the appropriate bandwidth management, we considered only the network bandwidth consumption that used a half of the total values. So, the bandwidth consumption per 1 call is about 8 KB/s for G.729 codec. From the experiment, when the concurrent calls reaches 10 calls, the

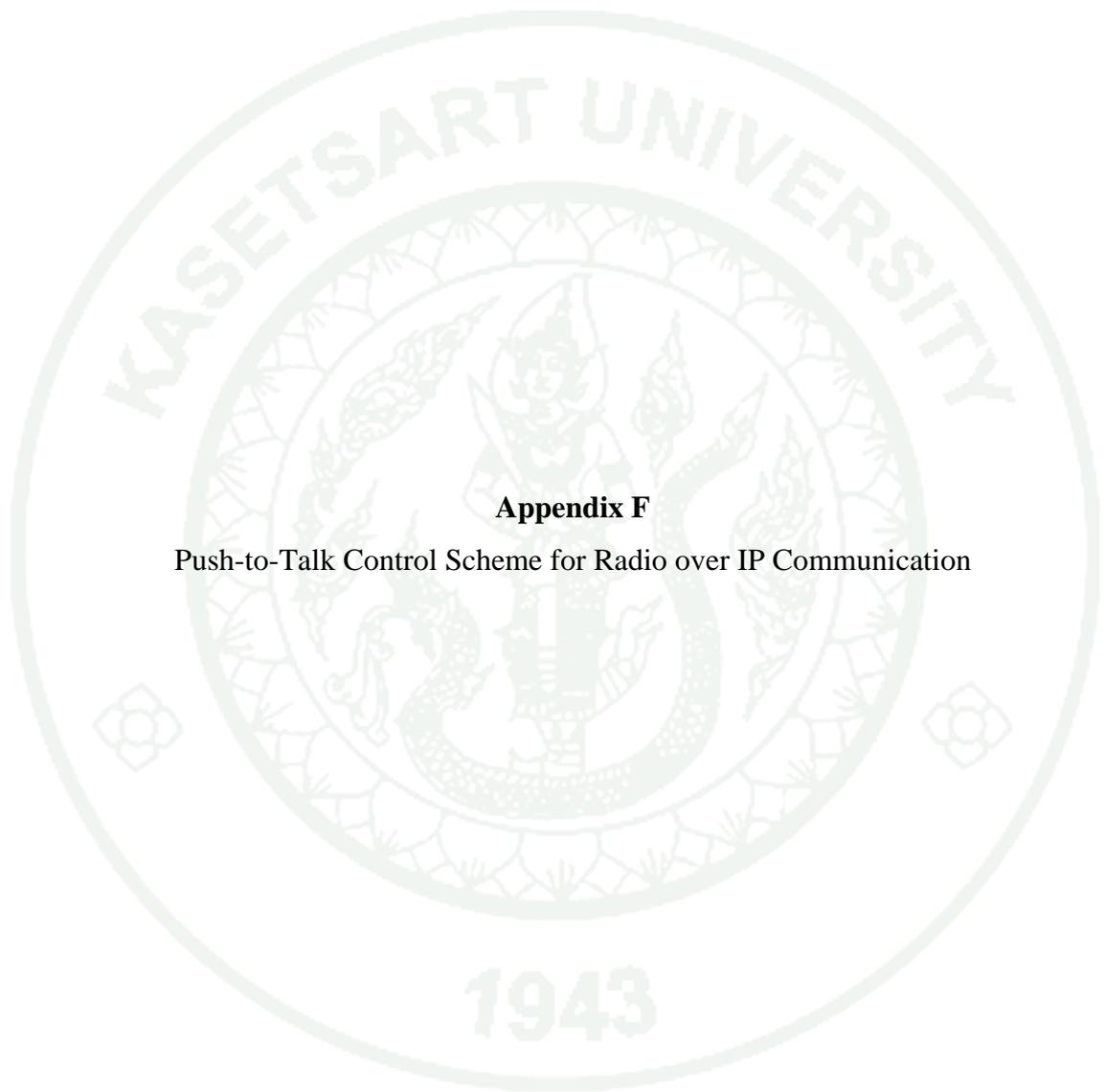
bandwidth consumption is about 63 kB/s(125.27 /2) or 0.50 Mb/s. We can determine that the voice transmission via 2 Mbps (2 downlink/1 uplink) IP-Star broadband using G.729 codec can support up to 20 calls (the bandwidth consumption of 20 calls is about 1000 kb/s or about 1 Mbps).

Table 3. Video Transmission via 2 Mbps IPstar Broadband satellite link using h.264 codec

Number of Call(s)	CPU%	Bandwidth (Receive/Transmit) KB/s
1	1%	113.97 / 134.83

Table 4. The summary of two codec coordination with no silent suppression) for voice and video transmission via 2 Mbps IPstar Broadband satellite link

codec	Device	Bandwidth		Jitter		Packet Loss	
		DL (kB/s)	UL (kB/s)	Fwd (ms)	Rev. (ms)	Fwd (%)	Rev. (%)
G.711	VoIP Phone	10.2	10.1	1.09	16.93	0.5	0.5
G.711	SoftPhone (voice)	9.0	9.8	0.98	12.94	16.3	16.3
G.711	SoftPhone (video)	23.0	49.4	1.19	19.67	0.9	0.9
G.711-G.729	VoIP Phone	3.5	3.3	0.8	11.96	7.9	7.9
G.711-G.729	SoftPhone (voice)	3.0	2.9	0.63	5.70	0	0
G.711-G.729	SoftPhone (video)	13.1	41.8	0.69	20.49	0.7	0.7
G.729	VoIP Phone	3.5	3.3	0.97	12.45	0.1	0.1
G.729	SoftPhone (voice)	3.0	2.9	1.16	12.22	0.1	0.1
G.729	SoftPhone (video)	13.1	41.8	1.30	33.52	0.7	0.7



Appendix F

Push-to-Talk Control Scheme for Radio over IP Communication

Push-to-Talk Control Scheme for Radio over IP Communication

S. Tabsombat¹, N. Pimpuch², A. Hiranya-ekapar², M. Raksapatcharawong², K. Yamaoka³, T. Phatrapornnant⁴, W. Phiwthongkham⁴, and P. Duangtanoo⁴

¹TAIST Tokyo Tech, ICTES Program, Department of Electrical Engineering, Kasetsart University, Bangkok, Thailand (tsirinya@hotmail.com)

²Department of Electrical Engineering, Kasetsart University, Bangkok, Thailand (fengmkr@ku.ac.th)

³Department of Communications and Integrated Systems, Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo, Japan (yamaoka.ss@titech.ac.jp)

⁴National Electronics and Computer Technology Center, Pathumthani, Thailand ({teera.phatrapornnant, waranyoo.phiwthongkham, phisanu.duangtanoo}@nectec.or.th)

ABSTRACT

Communication via IP can enable the interconnection between many different kinds of communication devices such as cell-phones, 2-way radio, PSTN, Internet phones, VoIP phones, etc. The important key of operating the radio communication or radio-phone hybrid communication is the Voice Operated eXchange (VOX) design by applying an auto-answer mode that supports the accurately enable and disable the voice transmission. In this paper, we present a control algorithm to form a high quality of VOX/PTT operation.

Keywords: Keywords- Radio Over IP, SIP, Push to talk, VOX

1. INTRODUCTION

Push To Talk (PTT) is the property of a walkie-talkie mobile radio. It used for allow the speaker to talk via device to group members by pressing the PTT botton. Since PTT is half duplex communication, only one group member is allowed to speak at a time, and the other members listen.

The traditional 2-way radio is not sustainable as convergence communication system technology. In RoIP communication, the system demands more than simple radio-to-radio communications. Some cellular telephones can allow PTT technologies. But, any alternative PTT communication supports the non-PTT telephone?

In this paper, we apply an auto-answer mode to radio-radio or radio-phone cooperation to manage SIP-based push-to-talk services for Radio over IP application.

The propose of the service is to incorporate the hybrid association of mobile radio devices, together with the telephones. With the implementation of Voice Operated eXchange (VOX) and Telephone Hybrid, telephone or any VoIP communication device can accept any incoming Push- to-talk sessions.

2. Radio over IP System Architecture

Radio over IP system is the transport of voice using packet-switched Internet Protocol (IP). By delivery voice from radio and convert to digital form, sent over the data network and then convert to voice signal to the destination. In this system, the core service platform is the SIP-based IP-PBX running under Linux OS[1].

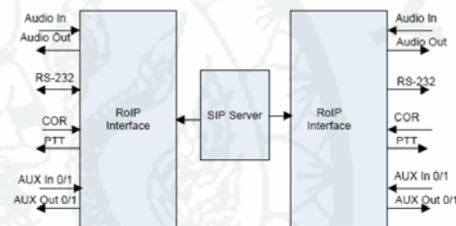


Figure 1. Block diagram of Radio over IP system

A PTT function allows transmission of the audio. Any time the user pushing the PTT button to speak on a hand held radio, voice will travel via radio waves to the other side end-user radio. The Carrier Operated Relay function (COR) is a receiver signal that gives a positive indication a carrier or signal is being received and the receiver is unscelched[2]. The device connected to the radio must have the ability to create the COR[2]. If the radio does not supply COR, the VOX (Voice Operated eXchange) canbe used to detect whenever the receiving radio voice signal exceeds a set threshold. The threshold can be varied for difference conditions.

In A/D and D/A conversion, the Analog Telephone Adaptor (ATA) devices are used, similar to the VoIP system. The difference is the RoIP system needs the Telephone to Radio Controller Circuit to support the Radio-Phone cooperation.

The controller circuit including with the MCU, Telephone Hybrid, Telephone Line Detection, Voice Detection, and Attenuator

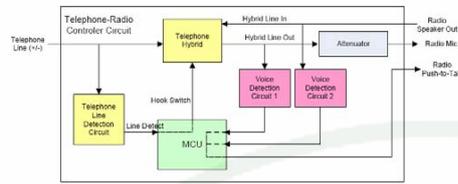


Figure 2. Block diagram of Telephone to Radio controller circuit

The most significant parts that perform the PTT control scheme consist of the Telephone Hybrid and Voice detection.

2.1 Telephone Hybrid

The Telephone Hybrid circuit is designed to transform the normal telephone line and separate to Line in and Line out form. Therefore, the communication between telephones and radios can be enabled.

2.2 Voice Detection

The Voice Detection circuit incorporates the necessary amplifiers, attenuators, level detectors, and control algorithm to form a high quality of VOX/PTT operation.

A VOX (Voice Operated eXchange) function is implemented in the Voice Detection Circuit using the VAD (Voice Activity Detector) feature. When a voice signal is detected at the radio port, an internal Carrier Operated Squelch (COS) signal is generated and transmitted to the destination. This will also enable the transmission of voice packets over the IP network. A hang period is automatically applied to the VOX algorithm. Full duplex operation is supported. On the other hand, the circuit is also designed for the voice transmission protection in order to disable the voice transmission while the receiving signal is detected.

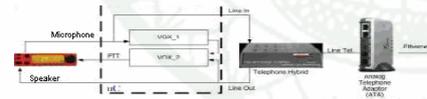


Figure 3. Radio-Phone cooperation via Telephone Hybrid circuit

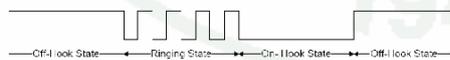


Figure 4. Example of telephone line diagram

For the telephone user, the Telephone Line Detection will detect the hook state signal and the ringing signal from the phone line. Then the microcontroller will compare the Line Out signal with the threshold voltage. If the Line Out signal > threshold, the VOX_2 will transmit the PTT signal to press the PTT. On the other hand, while the user from radio side is speaking, the VOX_1 will obstruct VOX_2 from transmitting signal to PTT by Looping back the signal to the Line Out.

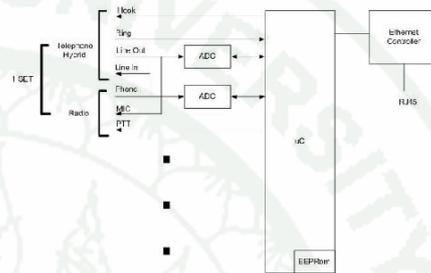


Figure 5. MCU unit diagram

3. PTT Control Scheme

Since the PTT is a one-way conversation, the control of talking scheme concerns with the priority management. There are many approaches to manage PTT support in PoC (Push-to-Talk over Cellular) based on SIP signaling. In [3], it adopts the RTP Control Protocol(RTCP) signaling to maintain the “token”. The token is carried in the RTCP signaling to exchange the talking privilege between PTT server and PTT group members. The other method in [4] use the SIP INFO and RTP session through the signaling message between the PTT server and PTT user. And in [5] using a context-aware rule for PTT service by filtering associated presence information to make a reasonable decision to adapt the PTT service. But they are not support the PTT service on regular phone over IP. In this paper, we apply the SIP server related to our previous study[1] which keeps compatible with SIP UAC, SIP UAS, and another SIP signaling. In pseudo diagram described below, the PTT signal is included.

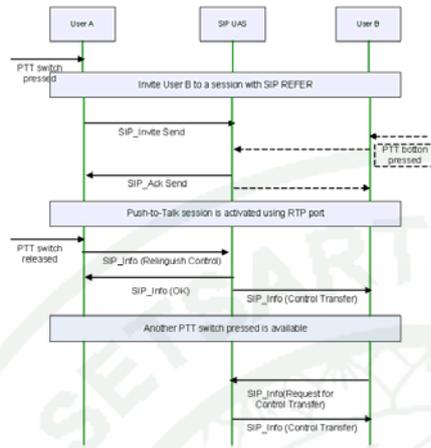


Figure 6. Signaling diagram of PTT control scheme

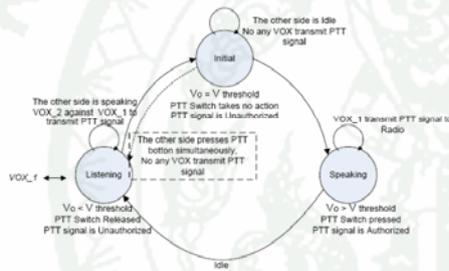


Figure 7. Single side of Push-to-Talk control scheme

State diagram in figure 7. describes the PTT performance over RoIP. For example, if one side (may be telephone) connected to VOX 1 and the other side (may be radio) connected to VOX 2, when the initial and speaking states are activated by the first side that connected to VOX 1, the VOX 1 will transmit the PTT signal to press the PTT, while looping back the signal to obstruct VOX 2 to transmit any signal. On the other hand, if the side that first reach the threshold is connected to VOX 2, the VOX 2 will transmit the PTT signal to press the PTT, as the same scheme but difference side. In case the both side press PTT simultaneously, no VOX can be sent any signal. In the experiment, where the PTT threshold was set to 5.0 volt, the end-to-end delay are shown in table 1.

TABLE 1. END-TO-END DELAY FROM PUSH-TO-TALK

Device	PTT Method	End to End Delay [s]
Radio to Radio	Push button key	2.0
Radio to Radio	VOX	1.0
Radio Telephone to Telephone	Push button key	2.0
Radio Telephone to Telephone	VOX	1.0
Radio to Radio	VOX	1.0

4. CONCLUSION

In this paper, we presented a call control scheme for PTT operation via RoIP system. By implementing the Voice Operated eXchange (VOX) function to making the priority of PTT call model and call control, in this scheme, the signaling message and voice data are encapsulated in a same data packet, so the call delay from the experiment result is based on the end-to-end delay. Using this service, we can enable the radio-phone hybrid communication using SIP and RTP/IP protocols.

ACKNOWLEDGMENT

This research is financially supported by Thailand Advanced Institute of Science and Technology - Tokyo Institute of Technology (TAIST-Tokyo Tech), National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology (Tokyo Tech) and Kasetsart University (KU). The authors would like to express their thanks to MICT (Ministry of Information and Communication Technology) for supporting this research through a consultancy project entitled "Prototyping a Communication System for Emergency Response via IP", grant no. 27/2552.

REFERENCES

[1] SirinyaTasombat, Monkol Raksapatcharawong, Katsunori Yamaoka, Teera Phatrapornmant, Waranyoo Phiwthongkham and Phisanu Duangtanoo, "Performance Analysis of Asterisk SIP Server for Emergency Communications", in Proceedings of the 2010 IEEE International Conference on Information and Communication Technology for Embedded Systems (IC-ICTES 2010), Bangkok, Thailand, 2010.

[2] Lin, J.-R., Pang, A.-C., and Wang, Y.-C., "iPTT: Peer-to-Peer Push-to-Talk for VoIP," Journal of

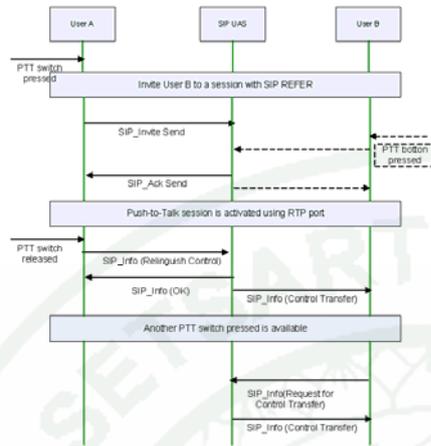


Figure 6. Signaling diagram of PTT control scheme

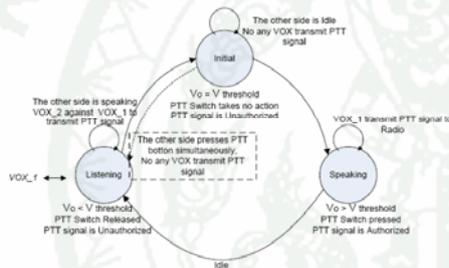


Figure 7. Single side of Push-to-Talk control scheme

State diagram in figure 7. describes the PTT performance over RoIP. For example, if one side (may be telephone) connected to VOX_1 and the other side (may be radio) connected to VOX_2, when the initial and speaking states are activated by the first side that connected to VOX_1, the VOX_1 will transmit the PTT signal to press the PTT, while looping back the signal to obstruct VOX_2 to transmit any signal. On the other hand, if the side that first reach the threshold is connected to VOX_2, the VOX_2 will transmit the PTT signal to press the PTT, as the same scheme but difference side. In case the both side press PTT simultaneously, no VOX can be sent any signal. In the experiment, where the PTT threshold was set to 5.0 volt, the end-to-end delay are shown in table 1.

TABLE 1. END-TO-END DELAY FROM PUSH-TO-TALK

Device	PTT Method	End to End Delay [s]
Radio to Radio	Push button key	2.0
Radio to Radio	VOX	1.0
Telephone to Telephone	Push button key	2.0
Radio to Telephone	VOX	1.0
Telephone to Radio	VOX	1.0

4. CONCLUSION

In this paper, we presented a call control scheme for PTT operation via RoIP system. By implementing the Voice Operated eXchange (VOX) function to making the priority of PTT call model and call control, in this scheme, the signaling message and voice data are encapsulated in a same data packet, so the call delay from the experiment result is based on the end-to-end delay. Using this service, we can enable the radio-phone hybrid communication using SIP and RTP/IP protocols.

ACKNOWLEDGMENT

This research is financially supported by Thailand Advanced Institute of Science and Technology - Tokyo Institute of Technology (TAIST-Tokyo Tech), National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology (Tokyo Tech) and Kasetsart University (KU). The authors would like to express their thanks to MICT (Ministry of Information and Communication Technology) for supporting this research through a consultancy project entitled "Prototyping a Communication System for Emergency Response via IP", grant no. 27/2552.

REFERENCES

[1] SirinyaTasombat, Monkol Raksapatcharawong, Katsunori Yamaoka, Teera Phatrapornmant, Waranyoo Phiwthongkham and Phisanu Duangtanoo, "Performance Analysis of Asterisk SIP Server for Emergency Communications", in Proceedings of the 2010 IEEE International Conference on Information and Communication Technology for Embedded Systems (IC-ICTES 2010), Bangkok, Thailand, 2010.

[2] Lin, J.-R., Pang, A.-C., and Wang, Y.-C., "iPTT: Peer-to-Peer Push-to-Talk for VoIP," Journal of

CURRICULUM VITAE

NAME : Ms.Sirinya Tabsombat

BIRTH DATE : December 27, 1978

BIRTH PLACE : Khon Kaen, Thailand

EDUCATION	: <u>YEAR</u>	<u>INSTITUTE</u>	<u>DEGREE/DIPLOMA</u>
	2008	Kasetsart Univ.	B.Eng. (Electrical)
	2010	Kasetsart Univ.	M.Eng. (ICTES)

POSITION/TITLE : Research Assistant

WORK PLACE : SCORPion Laboratory, Department of
Engineering, Kasetsart University

SCHOLARSHIP / AWARDS : TAIST Tokyo Tech Scholarship 2008-2010